uc3m | Universidad Carlos III de Madrid

Bachelor's Degree in
Telecommunication Technologies Engineering
Academic Year 2017/2018

Bachelor Thesis

"Generative Modeling using a database of patients with Acute Myeloid Leukemia"

Francisco José Cobo Celdrán

Supervised by:
Pablo Martínez Olmos
Leganés, June 2018

ABSTRACT

The main idea of this thesis is to apply unsupervised machine learning, particularly Generative Modeling to a database of patients with Acute Myeloid Leukemia. This approach allows discovering correlations among patients that doctors knew existed but did not know exactly how they worked. By selecting the right variables to study (the most important ones, which give the most information), the patients can be separated into several groups or *clusters*, containing those with similar characteristics. This is very interesting in medical data science (as well as in many other fields) because it enables to extract better, deeper and more interesting conclusions from the data. Generative Modeling is also a good approach when the amount of data is not enormous, the case of the database used for this thesis. The data is studied from an agnostic point of view, following a pure statistical analysis that is later double checked with experts on the field. A non-observed variable *(latent variable)* is assumed to explain the relations among the patients, and divide them into the mentioned clusters; using a proposed Gaussian-Bernoulli mixture model, that *latent variable* is inferred as a set of probabilities for each patient to belong in one of the defined clusters. The model is implemented in open-source programming language *Python*.

Key words: Unsupervised learning, Generative Modeling, Gaussian mixture, Bernoulli mixture, latent variable, data science, machine learning, Python.

RESUMEN EN CASTELLANO

La idea principal de este proyecto es aplicar aprendizaje máquina no supervisado, particularmente modelos generativos, a una base de datos de pacientes con leucemia mieloide aguda. Este análisis permite descubrir relaciones entre pacientes que los médicos conocían pero no sabían exáctamente cómo funcionaban. Seleccionando las variables correctas a analizar (aquellas que nos dan más información, las más significativas), los pacientes pueden ser separados en varios grupos, denominados *clusters*, cada uno conteniendo pacientes con características similares. Esto es especialmente interesante en la ciencia de datos médicos ya que permite obtener conclusiones muy importantes acerca de los datos empleados para esa separación. Los modelos generativos son también un buen método cuando la cantidad de datos disponible no es enorme, el caso de este trabajo. Los datos son estudiados desde un punto de vista agnóstico, siguiendo un análisis púramente estadístico que es comprobado con médicos expertos en el campo. Se asume que una variable no observable (denominada *variable latente*) explica las correlaciones entre los pacientes y los divide en los mencionados *clusters*; utilizando un modelo propuesto para una mezcla de Gausianas y Bernoullies, esa *variable latente* es inferida como un conjunto de probabilidades de que cada paciente pertenezca a una de las agrupaciones predefinidas. El modelo está implementado en el lenguaje de programación *open source* Python.

Palabras clave: Aprendizaje no supervisado, modelos generativos, mezcla de Gausianas, mezcla de Bernoullies, variable latente, ciencia de datos, aprendizaje máquina, Python.

DEDICATORIA Y AGRADECIMIENTOS

Este proyecto está dedicado a mi familia y amigos. A mis padres, por darme la oportunidad de hacer todo lo que he querido realizar a lo largo de mi vida, y particularmente de mis estudios; por apoyarme y confiar en mi. A mis hermanos por hacerme la vida mucho más divertida y por tener en ellos siempre compañía. A mis abuelos por su constante interés, por sus esfuerzos, por su confianza en mi. A mis tíos y primos por tantos y tan buenos recuerdos y momentos juntos. A mis amigos por todos los buenos momentos vividos y por todos los que nos quedan por delante, qué bueno ir cerrando etapas juntos, ¡qué ya van unas cuantas! También a toda la gente que se ha cruzado en mi etapa universitaria, especialmente a los compañeros y amigos de estos cuatro años tan duros e intensos, cómo nos lo hemos trabajado.

Me gustaría también agradecer a mi tutor del proyecto Pablo la oportunidad de realizarlo, he aprendido muchísimo y lo he disfrutado también. Agradecer también a la Universidad por el buen funcionamiento de la Escuela.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this first chapter, the motivation of the project, its goals and the personal learning outcomes will be described. Also, a brief description of the disease studied in the thesis will be presented. The project structure is also explained.

1.1. Motivation of the project

The main motivation for this project is to collaborate in the never-ending fight against cancer. This project was done in the framework of an ongoing collaboration with the Hematology Department of Madrid's Hospital Gregorio Marañón. The idea of the agreement is to put together a multidisciplinary team, doctors and engineers, seeking to get the best results by using the technical analysis done from the university side and the knowledge of the field from the doctors' side. The Hospital provided a quality database from 140 patients with Acute Myeloid Leukemia (AML), having different fields like age, gender, diagnosis date, leukocytes, platelets, blasts percentage and many others, like genetic mutations associated to the disease.

1.2. Goals

The goals of the project are the following:
- Finding several subsets of patients with similar qualities that lead to deeper understanding of how a particular genetic mutation or level of a particular enzyme affects the general survival of a patient, in order to find the best possible treatment.
- Analyzing how well the patients react to a particular treatment depending on their characteristics.
- Providing a framework, a generic Generative model implementation for Gaussian and binary data (studied as Bernoulli data), for its future use in other situations. This model is not yet implemented in any standard open-source library for Python and is available in my personal GitHub: github.com/franciscocobo/GaussianBernoulliMixtureModel.

1.3. Learning outcomes

During the development of this thesis, I have become aware of some of the most important challenges that data scientists face when dealing with real databases, in order to develop machine learning techniques. I have learned how important is the data curation when dealing with a database of real data, obtained and collected by humans. I have also understood that the outputs from an analysis might not be the ones expected, without that necessarily meaning to have obtained an incorrect result. It is also essential to have quality

data and to test the solutions proposed with structure-known data, generated by the tester. Particularly, I have become familiar with clustering algorithms, Generative models and the Expectation Maximization (EM) algorithm. I have learned concepts like scatter separability and Bayesian Information Criterion (BIC), useful in order to define the generative model for gaussian data. I have been part of a multidisciplinary research team, working with a shared-goal. Lastly, I have become familiar with the regulation regarding medical data uses for investigations, as well as the steps needed for an university project to be carried out successfully.

I got interested in data science during the third year of my bachelor degree, taking a couple of online courses. I was later proposed by my thesis supervisor the project and I thought that it was a very interesting and challenging topic, which would lead me to a first approach into the machine learning world. During my bachelor degree I have learned how to extract and understand complex information from books and technical documents, as well as coming up with solutions to different problems. I needed to put those skills to work during the development of the project.

1.4. Acute Myeloid Leukemia

Cancer begins when the cells in some part of the organism begin to grow out of control, sometimes spreading to other areas of the body [1]. There are many types of cancer. Leukemias are cancers that affect white blood cells. Acute means that it progresses rapidly and aggressively, and requires immediate treatment [2]. Acute Myeloid Leukemia (AML) refers not only to a single disease but rather to a group of leukemias that develop in the myeloid cell line in the bone marrow. Myeloid cells are red blood cells, platelets and white blood cells except for lymphocytes. AML occurs at any age but it is more common in adults over the age of 60, and also occurs more frequently in males than females. The most important factor when detecting and predicting AML is the genetic make-up of the leukaemic cells [3].

1.5. Project structure

The thesis begins with an explanation of what Generative Modeling is and why it is a good approach for the project, in Chapter 2. After that, in Chapter 3, the mixture models for Gaussian and binary data are developed step by step. Using those results, the model for this problem is proposed, a Gaussian-Bernoulli Mixture model (GBMM), Chapter 4. The mathematical derivation is explained and the model is implemented in Python. Next, toy examples with ground truth are provided for both the Gaussian Mixture model (GMM) and the Bernoulli Mixture model (BMM) in order to test that the implementation was correct, Chapter 5. Also, a toy example with ground truth is provided for the GBMM implementation. Several metrics are introduced in order to get interesting insights from the results. After having checked that the models work well, all three models are executed

with the real data (the patients database) in Chapter 6. There is also an analysis of the effect of the Transplant of Hemopoietic Progenitors (TPH), as it was requested by the Hospital, in Chapter 7. The results are presented and commented and also double checked with the doctors in Chapter 8. The regulatory framework of the project will be discussed in Chapter 9 and the socio-economic impact in Chapter 10. Finally, the conclusions of the project are developed in Chapter 11.

2. GENERATIVE MODELING

In this chapter, the state of the art in machine learning will be described. The specific type of machine learning approach followed in the project and the reason for it to be chosen will also be presented.

2.1. State of the art

It is very difficult to speak of *state of the art* in Machine Learning, as the growth is its technology so fast and quick. The hot topic nowadays is *Deep Learning*.

"Deep Learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example" [4]. Deep Learning techniques can achieve extremely high levels of accuracy, many times, exceeding human-level performance. These models are trained by using a gigantic set of labeled data and neural networks architectures that usually have many layers [4]. In the case of the database used for this thesis, the amount of data is quite small (about 140 samples) and the data is not labeled. This is the reason why the machine learning approach chosen for the project was a technique that seeks for statistical relations among the data, *Generative Modeling*.

2.2. Unsupervised machine learning

As the database used for the project contains data that is not labeled, and cannot be labeled until studied, the adopted approach was unsupervised learning. In particular, clustering techniques will be used to split the patients with AML in non-overlapping clusters (groups of data points with similar features). The simplest and most common clustering technique is called *k-means*, consisting on finding an assignment of data points to different clusters, as well as the centers of those clusters (vectors), such that the distances of each data point to its cluster center is a minimum [5]. This algorithm limits too much the kind of data that can be used with (for example, it does not have an appropriate solution with categorical data) and it is also non-robust to outliers.

2.3. Finite mixture modeling

"Mixture modelling is a popular approach for density estimation in both supervised and unsupervised pattern classification" [6]. These models are flexible enough for achieving a good trade-off between the model complexity (that is usually controlled by modifying the number of components of the mixture) and the amount of data available [7]. While the number of components varies, the parametric form for each of them remains the same.

Assuming that the data can be divided into non-overlapping sub-populations (not very realistic in many cases), the variable that identifies the groups is hidden. This can happen for many reasons, maybe it is difficult to collect (such as honest reporting of drug abuse), or perhaps it is inherently unobservable (such as high propensity to save money). This is the case where Finite Mixture models come in handy to model the probability of belonging to one of the groups. They are also useful in drawing inferences about how each group behaves and what are their main characteristics [8]. In the following chapters particular versions of mixture models will be introduced to tackle the study of the database used for the thesis.

2.4. Generative Modeling

There are two main approaches in machine learning [9]:

- Generative approach: the idea is to model the class-conditionals probability density functions (PDFs) and the prior probabilities. The model is called generative because the output of the algorithm explains how the data samples were generated according to the particular PDF and thus can be used to sample new data points. Examples of this approach are mixture of Gaussians, Bayesian networks...
- Discriminative approach: the posterior probabilities are directly estimated. There is no attempt to model the underlying probability distribution and thus new data points cannot be generated. It is more focused on the computational resources used, on the performance. Examples of the approach are Support Vector Machines (SVMs) or traditional neural networks.

As in this thesis the main purpose is to find interesting insights from the data, and being able to explain the characteristics of the patients grouped into each sub-population and the reason why each patient is in each subset; as well as being able to group a new patient into one of the predefined groups, Generative Modeling was the chosen approach.

3. MIXTURE MODELS AND EXPECTATION MAXIMIZATION

In this chapter, the theory behind general mixture models will be explained. The Expectation Maximization algorithm will be introduced as well as the specific derivations for Gaussian and Bernoulli mixture models.

3.1. Introduction

"Mixture models provide a framework for building complex probability distributions" [5]. A complex distribution $p(x)$ of an observed variable x is expressed in an easier and more tractable way by using a joint distribution over observed and *latent* (non-observed) variables: $p(x, z)$. Then, it is obtained the distribution of x alone by marginalizing that joint distribution: $p(x) = \Sigma_z p(x, z)$. In this way, latent variables allow complex distributions to be built from simpler components that can be analyzed easier. Also, mixture models allow treating data using *unsupervised learning*, "inferring a function to describe hidden structure from unlabeled data" [10]. This approach that is needed to be followed in this work. The purpose of the thesis will be to find *clusters* for the data set that are somehow medically interesting. It is understood that *clustering* consists on grouping objects into subsets such that objects withing each group share more common features with each other than with the objects outside of the group [11].

3.2. Expectation Maximization

Expectation Maximization (EM) is a general technique that aims to find ML (maximum likelihood) solutions (the parameters for that purpose) in the context of probabilistic models using *latent* variables. The main rough idea is to *maximize the probability of observing what was observed*. However, this process may lead to several problems and the complexity of finding the ML solution might be very high. This is why the process is divided into two steps, the E (Expectation) and the M (Maximization). Fitting the parameters of the mixture models is particularly challenging (it is needed to deal with non-convex problems), and so the EM algorithm provides a simple iterative solution that will always converge to a local minimum.

3.2.1. General EM

Given the joint distribution $p(X, Z|\theta)$, governed by the parameters θ (and being Z unobserved), the goal is to maximize the likelihood function $p(X|\theta)$ with respect to the parameters. As mentioned before, the computation of $p(X|\theta)$ is usually very complex, as well as the θ parameters that maximize it. EM provides an iterative solution for this

purpose [5]:

1) Choose an initial setting for the parameters θ^{old} (which can and should be initialized wisely so that the number of iterations is minimized).
2) **E step:** Evaluate $p(Z|X, \theta^{old})$. This function is the state of knowledge of the values of the *latent* variables. It is considered $\{X, Z\}$ as the complete data set and $\{X\}$ as the incomplete data set, the one that can be observed.
3) **M step:** Evaluate θ^{new}, considered by:

$$\theta^{new} = argmax_{\theta} Q(\theta, \theta^{old}) \tag{3.1}$$

where

$$Q(\theta, \theta^{old}) = \Sigma_Z p(Z|X, \theta^{old}) ln p(X, Z|\theta). \tag{3.2}$$

This is the expectation of the complete data *log likelihood*, evaluated for a general parameter θ. It works as the function to maximize in order to find the new parameters.
4) Check for convergence of either the *log likelihood* or the parameter values. EM algorithm will always give a higher likelihood solution. If the convergence criterion is not satisfied, then let:

$$\theta^{old} \leftarrow \theta^{new} \tag{3.3}$$

and return to step 2. This derivation is very useful in order to understand the process of EM and also to come back to it every time it is particularized it to a mixture.

3.2.2. Proof of EM

It will be proven the property that states that the EM algorithm always gives a solution in each iteration that has higher likelihood. In order to to this, it will be needed to make use of the *Jensen's Inequality* [12]:
If X is a random variable and φ is a convex function. Then $\varphi(E[X]) \leq E[\varphi(X)]$.

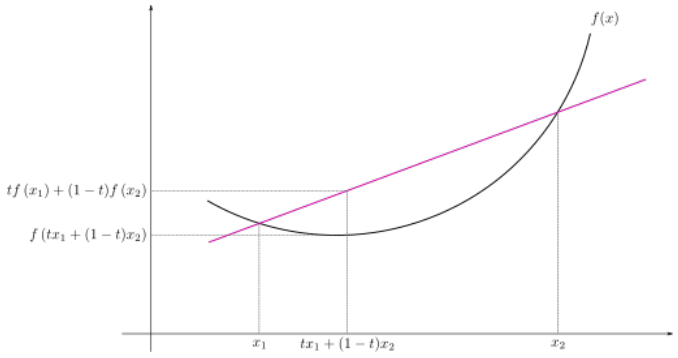


Fig. 3.1. Jensen's inequality [13]

Taking into consideration the latent variable Z, the *log likelihood* can be written as:

$$ln\{p(X|\theta)\} = ln\{\Sigma_Z p(X, Z|\theta)\} \tag{3.4}$$

That can be also maximized, taking into consideration the EM steps previously defined. Now the *Jensen's Inequality* [12] will be used to transform the *log likelihood* 3.4 function. The function is re-written by mutipliying and dividing by $q(Z)$, that represents an arbitrary distribution for the random variable Z:

$$\begin{aligned} ln\{p(X|\theta)\} &= ln\left\{\Sigma_Z \frac{q(Z)p(X,Z|\theta)}{q(Z)}\right\} \\ &= ln\,\mathbb{E}_q\left\{\frac{p(X,Z|\theta)}{q(Z)}\right\} \\ &\geq \mathbb{E}_q\left\{ln\frac{p(X,Z|\theta)}{q(Z)}\right\} \\ &\geq \Sigma_Z q(Z) ln\left\{\frac{p(X,Z|\theta)}{q(Z)}\right\} \\ &= \Sigma_Z q(Z) ln\{p(X,Z|\theta)\} - \Sigma_Z q(Z) ln\{q(Z)\} \end{aligned} \tag{3.5}$$

Therefore, if we let $q(Z) = p(Z|X,\theta^{old})$, then:

$$\begin{aligned} ln\{p(X|\theta^{new})\} &\geq \Sigma_Z q(Z) ln\{p(X,Z|\theta^{new})\} - \Sigma_Z q(Z) ln\{q(Z)\} \\ &= Q(\theta,\theta^{new}) - \Sigma_Z q(Z) ln\{q(Z)\} \\ &\geq Q(\theta,\theta^{old}) - \Sigma_Z q(Z) ln\{q(Z)\} \\ &= ln\{p(X|\theta^{old})\} \end{aligned} \tag{3.6}$$

Which proves that the EM algorithm will converge to a ML solution of the problem, as it always outputs a higher log likelihood function.

Databases have heterogeneous attributes and that forces the creation of models dealing with different PDFs, like Gaussian and Bernoulli.

3.3. Gaussian mixture model

3.3.1. Introduction

This section aims to explain in an easy and clear way the application of the EM algorithm, to the GMM (Gaussian mixture model). Information from [5] and [12], developing longer derivations of the equations in order to clarify the results.

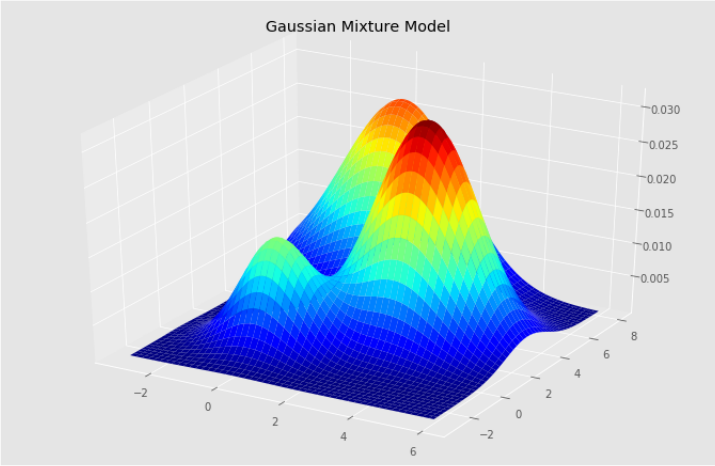


Fig. 3.2. Gaussian mixture example plot

A Gaussian distribution is defined as follows:

$$\mathcal{N}(X|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma|^{1/2}}exp\{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\} \tag{3.7}$$

The GMM distribution is a linear superposition of Gaussians:

$$p(X) = \Sigma_{k=1}^{K}\pi_k\mathcal{N}(X|\mu_k,\Sigma_k) \tag{3.8}$$

Subject to (as it has to fulfill the properties of PDFs):

$$\Sigma_{k=1}^{K}\pi_k = 1 \tag{3.9}$$

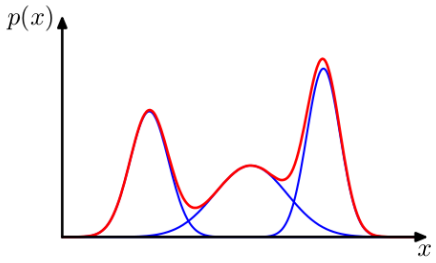


Fig. 3.3. Gaussian mixture example graph, [5]

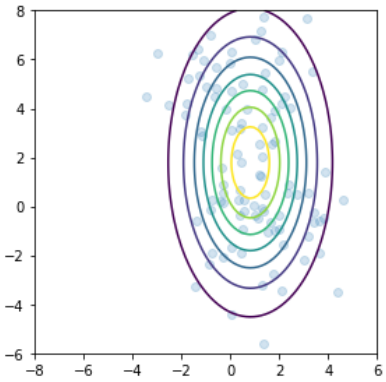
So, for the GMM the following parameters are introduced:

K, the **number of Guassian components**
$\pi_1...\pi_k$, the mixture **weights** of the components
$\mu_1...\mu_k$, the **mean** of each component
$\Sigma_1...\Sigma_k$, the **covariance matrix** of each component

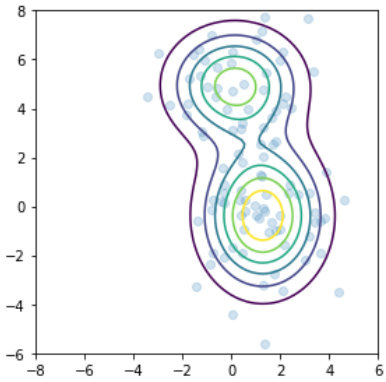
Using them, the mixture is completely defined and thus samples can be generated (s_1, $s_2...s_n$) from the distribution.

3.3.2. What is the reason to use GMM?

It provides a "richer class of density models that the single gaussian" [5]. Data can be better described using several gaussians rather than a single one, as it is seen in the example below.



(a) Single Gaussian



(b) Gaussian mixture of 2 components

Fig. 3.4. Why GMM?

3.3.3. How to apply EM algorithm to GMM

As it was mentioned before, in order to simplify the computations, the notion of *latent variables* is introduced. *Latent variables* are understood as "variables that are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed (directly measured)" [5].
The goal of the EM algorithm is to find ML solutions of the observed data, based on some set of parameters θ:

$$max_\theta\{p(X|\theta)\} \tag{3.10}$$

for models having latent variables:

$$ln\{p(X|\theta)\} = ln\{\Sigma_Z p(X,Z|\theta)\} \tag{3.11}$$

As, being the *ln* an only-increasing function, it can maximized both the function and the *ln* of it. This, again, will simplify the calculus. Note that this be could be applied to continuous latent variables just by replacing the sum over Z with an integral [5].

Introduction of the latent variable in GMM

Given the GMM, it is introduced a K-dimensional binary random variable Z, which only takes value 1 at the element z_k while all the other elements of the array are equal to 0: $Z = (0, 0, ..., 1, 0, ..., 0)$
There are therefore K possible states for Z. It is defined:

$$p(z_k = 1) = \pi_k \tag{3.12}$$

Which can be understood as a *mixing coefficient* for the k^{th} component of the mixture. It will be later seen that this coefficient will tell the average *responsibility* that each component takes for explaining the data points.
As each element is considered independent of each other, the likelihood of the latent variable Z can be defined as:

$$p(Z) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{3.13}$$

It is important to take a moment to understand this formula as it will be of great importance in the following calculations. The product is written in a very particular way, that will ease future math computations as the exponent will be transformed in a product by taking the *ln*. The exponent weights the terms of the product with either an unit value (when $z_k = 0$) or the actual value of the likelihood.

Coming back to the GMM, the conditional likelihood of the observed data X given a particular z_k can be defined as a Gaussian:

$$p(X|z_k = 1) = \mathcal{N}(X|\mu_k, \Sigma_k) \tag{3.14}$$

Generalizing the previous equation:

$$p(X|Z) = \prod_{k=1}^{K} \mathcal{N}(X|\mu_k, \Sigma_k)^{z_k} \tag{3.15}$$

And now, it can easily computed (using the *Law of total probability*) the marginal distribution of the observed data X using $p(Z)$ (3.13) and $p(X|Z)$ (3.15):

$$p(X) = \Sigma_Z p(X|Z)p(Z) = \Sigma_Z \prod_{k=1}^{K} \mathcal{N}(X|\mu_k, \Sigma_k)^{z_k} \prod_{k=1}^{K} \pi_k^{z_k} = \Sigma_{k=1}^{K} \pi_k \mathcal{N}(X|\mu_k, \Sigma_k) \tag{3.16}$$

Again, take intro consideration that substituting the sum with an integral, the result for continuous latent variables is obtained.

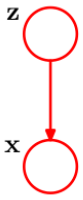


Fig. 3.5. Graphical model Z, [5]

Now we can compute $p(Z|X)$, using $p(X, Z) = p(X|Z)p(Z)$ (Bayes). $p(Z|X)$ is the posterior probability of the latent variable Z, needed for the development of the EM algorithm (section: 3.2.1). $p(Z|X)$ is also usually referred as the *responsibility* (notion mentioned before):

$$\gamma(z_k) = p(z_k = 1|X) = \frac{p(X|z_k = 1)p(z_k = 1)}{p(X)} = \frac{\mathcal{N}(X|\mu_k, \Sigma_k)\pi_k}{\Sigma_{j=1}^{K} \pi_j \mathcal{N}(X|\mu_j, \Sigma_j)} \tag{3.17}$$

$\gamma(z_{n,k})$ being the responsibility of the component k^{th} for explaining the data point x_n. This will be the prediction of Z, the set of probabilities for each of the data points to belong to the pre-defined *clusters*.

$$\gamma(z_{n,k}) = p(z_k = 1|x_n) = \frac{\mathcal{N}(x_n|\mu_k, \Sigma_k)\pi_k}{\Sigma_{j=1}^{K}\pi_j\mathcal{N}(x_n|\mu_j, \Sigma_j)} \tag{3.18}$$

Finding the ML solution

The whole purpose is to find the ML solution of the problem. That means that the likelihood of the observed data, $p(X)$, should be maximized. This will be done with respect to some parameters that, in the particular case of the GMM are, as mentioned before:

The **mean** of each Gaussian component: μ_k.
The **covariance matrix** of each Gaussian component: Σ_k.
The **mixing coefficients:** $\pi_k = p(z_k = 1)$.

Having a data set of observations $\{x_1, ..., x_N\}$ and wanting to model it using a Gaussian mixture model, the data could be represented by a NxD matrix, being N the number of data vectors and D the dimension of the vector; being the *columns* of different types of data it contains (the different features). Taking into consideration the parameters that the model depends on, the likelihood function can be written as:

$$p(X|\pi, \mu, \Sigma) = \Sigma_{k=1}^{K} p(X, Z|\pi_k, \mu_k, \Sigma_k) = \Sigma_{k=1}^{K}\pi_k\mathcal{N}(X|\mu_k, \Sigma_k) \tag{3.19}$$

And thus the log likelihood, that can also be maximized. Taking all the data vectors (N):

$$\begin{aligned} ln\{p(X|\pi, \mu, \Sigma)\} &= \Sigma_{n=1}^{N} ln\{\Sigma_{k=1}^{K} p(X, Z|\pi_k, \mu_k, \Sigma_k)\} \\ &= \Sigma_{n=1}^{N} ln\{\Sigma_{k=1}^{K}\pi_k\mathcal{N}(x_n|\mu_k, \Sigma_k)\} \end{aligned} \tag{3.20}$$

We find the ML solution taking the derivatives of $lnp(X|\theta)$ (equation 3.20) to *zero* with respect to μ_k, Σ_k and π_k.

Firstly with μ_k:

$$0 = -\Sigma_{n=1}^{N} \frac{\pi_k\mathcal{N}(x_n|\mu_k, \Sigma_k)}{\Sigma_{j=1}^{K}\pi_j\mathcal{N}(x_n|\mu_j, \Sigma_j)}\Sigma_k(x_n - \mu_k)$$

It is easily seen that the first term of the multiplication inside $\Sigma_{n=1}^{N}$ equals the posterior probability of the latent variable, or responsibility $\gamma(z_{n,k})$ (equation 3.18).
Thus:

$$0 = -\Sigma_{n=1}^{N}\gamma(z_{n,k})\Sigma_k(x_n - \mu_k)$$

Assuming the covariance matrix Σ_k to be non-singular (invertible), both sides can multiplied by $(\Sigma_k)^{-1}$, obtaining:

$$\mu_k = \frac{\Sigma_{n=1}^{N} \gamma(z_{n,k}) x_n}{\Sigma_{n=1}^{N} \gamma(z_{n,k})}$$

In the implementation of the algorithm it will be needed to see how the non-singular matrices problem is approached. Defining:

$$N_k = \Sigma_{n=1}^{N} \gamma(z_{n,k}) \tag{3.21}$$

as the effective number of points assigned to cluster k [5], which makes sense by the definition of $\gamma(z_{n,k})$ given in equation 3.18, it can be written:

$$\mu_k = \frac{1}{N_k} \Sigma_{n=1}^{N} \gamma(z_{n,k}) x_n \tag{3.22}$$

This definition of the mean for the k^{th} Gaussian component makes a lot of sense as it is computing the weighted mean of all of the points in the data set X, in which the weightening factor is given by the posterior probability $\gamma(z_{n,k})$, that tells how each component k was responsible for generating the data point x_n [5].

Secondly, Σ_k. Setting the derivative of equation 3.19 to *zero* with respect to Σ_k, and making use of the result of the ML solution for a single gaussian (equation 3.7), which can be found in section 2.3.4 of [5]:

$$\Sigma_{ML} = \frac{1}{N} \Sigma_{n=1}^{N} (x_n - \mu_{ML})(x_n - \mu_{ML})^T$$

$$E[\mu_{ML}] = \mu$$

$$E[\Sigma_{ML}] = \frac{N-1}{N} \Sigma$$

$$\Sigma = \frac{1}{N-1} \Sigma_{n=1}^{N} (x_n - \mu_{ML})(x_n - \mu_{ML})^T$$

it can easily be obtained:

$$\Sigma_k = \frac{1}{N_k} \gamma(z_{n,k})(x_n - \mu_k)(x_n - \mu_k)^T \tag{3.23}$$

And lastly, the *mixing coefficient* π_k. However, it is important to take the constraint 3.9: $\Sigma_{k=1}^{K} \pi_k = 1$ into consideration. This is done by using the Lagrange multiplier [12].

Lagrange multiplier

It is needed to maximize $f(x, y)$ subject $g(x, y) = c$.
Let $\Lambda(x, y, \lambda) = f(x, y) + \lambda(g(x, y) - c)$.
Then if (x_0, y_0) is a maximum of the original f, there exists (x_0, y_0, λ_0) that is a stationary point for the Λ function.

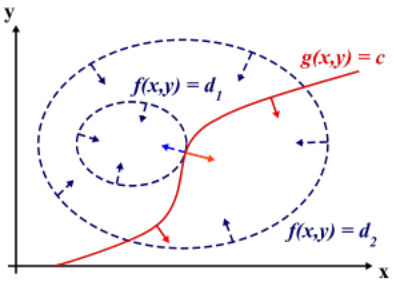


Fig. 3.6. Lagrange multiplier, [12]

"The contour lines of f and g touch when the tangent vectors of the contour lines are parallel. Since the gradient of a function is perpendicular to the contour lines, this is the same as saying that the gradients of f and g are parallel" [13].
So $\nabla_{x,y} f = -\lambda \nabla_{x,y} g$.
Combining with the constraint, $\nabla_{x,y,\lambda} \Lambda = 0$
Applying this to the problem, the following function needs to be maximized:

$$lnp(X|\pi, \mu, \Sigma) + \lambda(\Sigma_{k=1}^{K} \pi_k - 1) \tag{3.24}$$

Maximizing it with respect to π_k:

$$0 = \Sigma_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\Sigma_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} + \lambda$$

Where again the responsibilities appear:

$$0 = \Sigma_{n=1}^{N} \gamma(z_{n,k}) + \lambda$$

Taking into consideration the definition for the effective number of points assigned to cluster k (equation 3.21), [5]:

$$0 = N_k + \lambda$$

Mutiplying both sides by π_k:

$$0 = \pi_k N_k + \pi_k \lambda$$

Applying constraint 3.9, and summing over k:

$$\lambda = -N$$

So:

$$\pi_k = \frac{\Sigma_{n=1}^{N} \gamma(z_{n,k})}{N} = \frac{N_k}{N} \tag{3.25}$$

Which, again, makes a lot of sense because the *mixing coefficient* for the k^{th} component is given by the average responsibility which that component takes for explaining the distribution of the data points.

At this point it is needed to emphasize that the solutions for μ_k, Σ_k and π_k found in equations 3.22, 3.23, and 3.25, are not close-form solutions for the parameters of the mixture model because the responsibilities $\gamma(z_{n,k})$ depend on the parameters through equation 3.18, [5]. However, they are valid in this case in order to define a simple iterative solution for the EM algorithm.

3.3.4. Iterative solution for EM in GMM

The above process can be summarized in these steps, similar to the ones in section: 3.2.1.
1) Initialize the means μ_k (3.22), covariances Σ_k (3.23), and mixing coefficients π_k (3.25). This is sometimes done using *k-means*. Also evaluate the value of the *log likelihood* (equation 3.20).
2) **E step** Evaluate the posterior probabilites, the responsibilities, $\gamma(z_{n,k})$, (equation 3.18) using the current parameter values.
3) **M step** Re-estimate the parameteres using the current responsibilites. Using the formulas below the ML solution can be found:

$$\mu_k^{new} = \frac{1}{N_k} \Sigma_{n=1}^{N} \gamma(z_{n,k}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \gamma(z_{n,k})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

Being N_k defined in equation 3.21.
4) Evaluate the *log likelihood* (3.20) to check for convergence, if not, return to *step 2*.

3.3.5. Python implementation

Implemenation using open source libraries like Numpy [14], Scipy [15], Pandas [16].

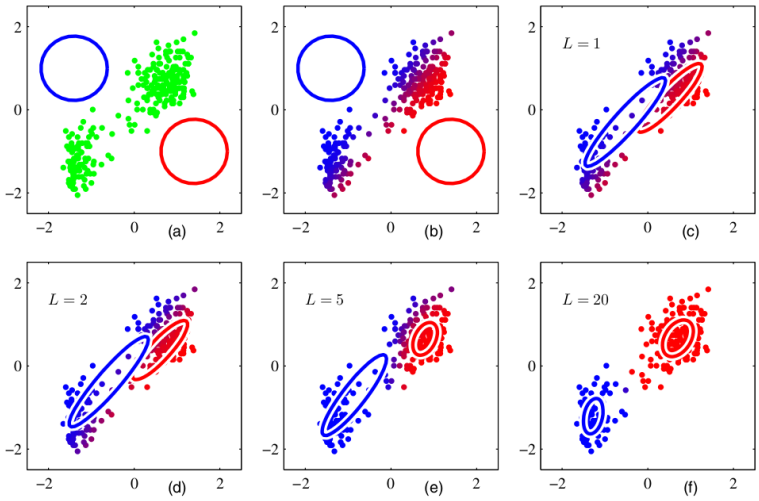


Fig. 3.7. Example of 20 iterations of EM algorithm in a GMM [5]

3.4. Mixture of Bernoulli distributions

3.4.1. Introduction

In the previous section it was developed the EM algorithm for continuous variables (those particularly being Gaussians). Now, it is proposed to apply EM to mixture of discrete variables, particularly discrete binary variables: Bernoulli mixture models (BMM). Derivations from [5] and [17].

Bernoulli distribution

The Bernoulli distribution is the probability distribution of a random variable that takes the value 1 with probability α and the value 0 with probability $q = 1 - \alpha$. The most clear and simple example is tossing a coin [18]. This would be the case for a single Bernoulli:

$$p(X|\alpha) = \alpha^X(1-\alpha)^{1-X} \tag{3.26}$$

where $X = 0, 1$ (head/tails).

Mixture of two Bernoullis

If instead of tossing one coin, it is done with two, the mixture distribution would be:

$$p(X|\alpha,\pi) = \pi_1\alpha_1^X(1-\alpha_1)^{1-X} + \pi_2\alpha_2^X(1-\alpha_2)^{1-X} \tag{3.27}$$

with parameters $\alpha = [\alpha_1, \alpha_2]$ choosing coins with probabilities $\pi = [\pi_1, \pi_2]$.

BMM

Considering now the general Bernoulli mixture model. Let D be a set of independent binary variables, that will be considered as $x_i, i, ..., D$. Each of them follows a Bernoulli distribution with parameter α_i:

$$p(X|\alpha) = \prod_{i=1}^{D} \alpha_i^{x_i}(1-\alpha_i)^{(1-x_i)} \tag{3.28}$$

where $x = (x_1, ..., x_D)^T$ and $\alpha = (\alpha_1, ..., \alpha_D)^T$. Following the example given before, this would be a set of D coins. The mean and covariance of this distribution are:

$$E[X] = \alpha \tag{3.29}$$

$$cov[X] = diag\{\alpha(1-\alpha)\} \tag{3.30}$$

Considering now a finite mixture of these distributions:

$$p(X|\alpha, \pi) = \sum_{k=1}^{K} \pi_k p(X|\alpha_k) \tag{3.31}$$

where $\alpha = [\alpha_1, ..., \alpha_k]$ and $\pi = [\pi_1, ..., \pi_k]$ (*mixing coefficient*), subject to $\sum_k \pi_k = 1$, and:

$$p(X|\alpha_k) = \prod_{i=1}^{D} \alpha_{k,i}^{x_i}(1-\alpha_{k,i})^{(1-x_i)} \tag{3.32}$$

Again, this distribution could remind to K bags of D coins, where each bag k is chosen with probability π_k [18]. It tells the probability that X is generated by a cluster k. Mean and covariance are:

$$E[X] = \sum_{k=1}^{K} \pi_k \alpha_k \tag{3.33}$$

$$cov[X] = \sum_{k=1}^{K} \pi_k\{\Sigma_k + \alpha_k\alpha_k^T\} - E[X]E[X]^T \tag{3.34}$$

where $\Sigma_k = diag\{\alpha_k(1-\alpha_k)\}$ Now, the matrix covariance $cov[X]$ is no longer diagonal and so the mixture will capture the correlation among the variables. Having in mind the EM algorithm, computing the log likelihood of 3.31 (the observed data), the result for the model will be:

$$lnp(X|\alpha, \pi) = \sum_{n=1}^{N} ln\left\{\sum_{k=1}^{K} \pi_k p(x_n|\alpha_k)\right\} \tag{3.35}$$

being $X = [x_1, ..., x_N]$, the observed data set. The first Σ_n in the formula is due to the logarithm and the second Σ_k is due to the mixture.

3.4.2. How to apply EM to BMM

In order to apply the EM algorithm to the model that has just being defined, it will again be introduced the notion of *latent variables*. It is defined an unobserved variable Z associated with each instance of X. Z is again defined as a binary *K-dimensional* variable having a single component equal to 1 and all other components equal to 0 (indicating the *cluster* that x_n belongs to). $Z = [z_1, ..., z_K]^T : Z = [0, 0, 1, ..., 0]^T$. Obviously: $\Sigma_{k=1}^{K} \pi_k = 1$. As $p(z_k = 1) = \pi_k$, then:

$$p(Z|\pi) = \prod_{k=1}^{K} \pi_k{}^{z_k} \tag{3.36}$$

than defines the prior distribution for the *latent variables*. In the same way, defining $p(x|z_k = 1) = p(x|\alpha_k)$, then it can be defined:

$$p(X|Z, \alpha, \pi) = \prod_{k=1}^{K} p(X|\alpha_k)^{z_k} \tag{3.37}$$

the conditional distribution of X.
Combining all the K *latent variables* into a set $Z = [z_1, ..., z_K]$:

$$p(Z|\alpha, \pi) = \prod_{k=1}^{K} p(z_k|\pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{n,k}} \tag{3.38}$$

and in the same way, by defining $X = [x_1, ..., x_N]$, the observed variables distribution (the available data) can be defined as:

$$\begin{aligned} p(X|Z, \alpha, \pi) &= \prod_{n=1}^{N} p(x_n|z_n, \alpha, \pi) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} p(x_n|\alpha_k)^{z_{n,k}} \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\prod_{i=1}^{D} \alpha_{k,i}^{x_{n,i}} (1 - \alpha_{k,i})^{(1-x_{n,i})} \right)^{z_{n,k}} \end{aligned} \tag{3.39}$$

using the definition in 3.32. And now it can be defined the joint probability as $p(X, Z|\alpha, \pi) = p(X|Z, \alpha, \pi)p(Z|\alpha, \pi)$:

$$p(X, Z|\alpha, \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_k \prod_{i=1}^{D} \alpha_{k,i}^{x_{n,i}} (1 - \alpha_{k,i})^{(1-x_{n,i})} \right)^{z_{n,k}} \tag{3.40}$$

this is the likelihood is aimed to maximize, so computing its natural logarithm for easier computation:

$$ln p(X, Z|\alpha, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{n,k} \left(ln\pi_k + ln \sum_{i=1}^{D} x_{n,i} ln\alpha_{k,i} + (1 - x_{k,i}) ln(1 - \alpha_{k,i}) \right) \tag{3.41}$$

For defining the E step it is needed $p(Z|X,\theta)$. This will be done, again, using *Bayes*: $p(x,z) = p(x|z)p(z)$. It will be defined, as it was done in the GMM case, the posterior probability or *responsibility* that each component takes for explaining the data.

$$\begin{aligned}\gamma(z_{n,k}) &= p(z_{n,k}=1|x) = E[z_{n,k}] \\ &= \frac{p(x|z_{n,k}=1)p(z_{n,k}=1)}{p(x)} \\ &= \frac{p(x_n|\alpha_k)\pi_k}{\Sigma_{j=1}^{K}\pi_j p(x_n|\alpha_j)} \\ &= \frac{\pi_k \prod_{i=1}^{D} \alpha_{k,i}^{x_{n,i}}(1-\alpha_{k,i})^{(1-x_{n,i})}}{\Sigma_{j=1}^{K}\pi_j \prod_{i=1}^{D} \alpha_{j,i}^{x_{n,i}}(1-\alpha_{j,i})^{(1-x_{n,i})}}\end{aligned} \tag{3.42}$$

that will be the function that is evaluated in the E step of the EM algorithm.

In the M step the parameters are updated (in this case α_k and π_k) in order to maximize the function $Q(\theta,\theta^{old}) = \Sigma_Z p(Z|X,\theta^{old}) ln p(X,Z|\theta)$ in BMM defined as:

$$Q(\theta,\theta^{old}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})\left(ln\pi_k + ln\sum_{i=1}^{D} x_{n,i} ln\alpha_{k,i} + (1-x_{k,i})ln(1-\alpha_{k,i})\right) \tag{3.43}$$

Firstly, maximizing with respect to α_k 3.43, by setting its derivative to *zero*:

$$\begin{aligned}&\sum_{n=1}^{N}\gamma(z_{n,k})\sum_{i=1}^{D}\left(\frac{x_{n,i}}{\alpha_{k,i}} - \frac{1-x_{k,i}}{1-\alpha_{k,i}}\right) \\ &= \sum_{n=1}^{N}\gamma(z_{n,k})\sum_{i=1}^{D}\left(\frac{x_{n,i}-\alpha_{k,i}}{\alpha_{k,i}(1-\alpha_{k,i})}\right) \\ &= \sum_{n=1}^{N}\gamma(z_{n,k})\left(\frac{x_n-\alpha_k}{\alpha_k(1-\alpha_k)}\right) = 0\end{aligned} \tag{3.44}$$

Defining:

$$N_k = \sum_{n=1}^{N}\gamma(z_{n,k}) \tag{3.45}$$

as the number of data points associated with component k [5], and

$$\overline{x}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{n,k})x_n \tag{3.46}$$

as the weighted mean of the data, with weighting coefficients given by the responsibilities that component k takes for data points [5], it can be concluded from 3.44:

$$\alpha_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{n,k})x_n = \overline{x}_k \tag{3.47}$$

Secondly, maximizing 3.43 with respect to the mixing coefficients π_k. Again, it will be needed to proceed using the *Lagrange multiplier*, getting following function to maximize:

$$Q(\theta,\theta^{old}) + \lambda(\Sigma_{k=1}^{K}\pi_k - 1) \tag{3.48}$$

Being $Q(\theta, \theta^{old})$ the one defined in 3.43. Setting derivatives with respect to π_k:

$$\frac{1}{\pi_k} \sum_{n=1}^{N} \gamma(z_{n,k}) + \lambda = 0 \tag{3.49}$$

that yields to:

$$\pi_k = \frac{-N_k}{\lambda} \tag{3.50}$$

Following the same procedure that was done for this part in GMM, $\lambda = -N$, and thus:

$$\pi_k = \frac{N_k}{N} \tag{3.51}$$

Again, this definition makes a lot of sense as the mixing coefficient for component k is given by the number of points in the data set explained by that component.

3.4.3. Iterative solution for EM in BMM

The above process can be summarized in these steps, similar to the ones in section: 3.2.1.
1) Initialize the parameters α_k, π_k.
2) **E step**: evaluate $\gamma(z_{n,k})$ (3.42) using the current parameter values.
3) **M step**: update the parameneters:

$$\alpha_k^{new} = \overline{x}_k \tag{3.52}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{3.53}$$

where $\overline{x}_k = \frac{1}{N_k} \Sigma_{n=1}^{N} \gamma(z_{n,k}) x_n$ and $N_k = \Sigma_{n=1}^{N} \gamma(z_{n,k})$.

4) Evaluate the *log likelihood* (3.41) to check for convergence, if not, return to *step 2*.

3.4.4. Python Implementation

Implemenation using open source libraries like Numpy [14], Scipy [15], Pandas [16].

3.5. Summarizing of the idea of the mixture model

The main idea for the development of this section is that, in a given data set, there exist a hidden variable that explains the correations among the different data points (having these a set of features that identify them). This correlation consists on clusters of those data points, grouping the ones that have similar features. Mixture models induce that hidden variable that cannot be observed by giving a function that has been defined as $\gamma(z_{n,k})$, that tells the probability of each data point to belong to a particular cluster.

4. MODEL PROPOSED FOR THIS PROJECT: GAUSSIAN-BERNOULLI MIXTURE

In this chapter, the particular mixture model proposed for dealing with the database used in the thesis will be described.

4.1. Introduction

As in the data set used in this thesis there existed both real and binary data, as well as categorical data that was treated as binary, it was needed a model that could combine both types of data in order to get the most information possible out of the data set. This is the reason why the following model is proposed, an EM implementation of a combination of Bernoullis and Gaussians. Features for each data point containing real and binary data are separated but are not analysed independently.

4.2. Derivation

In the GBMM (Gaussian Bernoulli mixture model) is needed to define a vector of parameters θ that includes both the parameters of the GMM and the BMM, as well as the mixing coefficients π that now works for both distributions. It is defined $\theta = [\theta_{gauss}, \theta_{ber}, \pi]$, $\pi = [\pi_1, ..., \pi_K]$ (being K the number of clusters), $\theta_{gauss} = [\mu_{1,1}, ..., \mu_{K,G}, \Sigma_1, ..., \Sigma_G]$ (being G the number of Gaussian features), $\theta_{ber} = [\alpha_{1,1}, ..., \alpha_{K,B}]$ (being B the number of Bernoulli features). Also, it is denoted X as the Gaussian observations and Y as the Bernoulli observations, so that $f(X)$ is the density function of Gaussian data, $f(Y)$ is the density function of Bernoulli data and $O = [X, Y]$.

GBMM is a joint mixture model that assumes that for each component, data of the two distributions are independent.
- For Gaussian features, each component i is assumed to be a distribution in the GMM:

$$f(X|\theta_{gauss,i}) = \frac{1}{(2\pi)^{G/2}} \frac{1}{|\Sigma_i|^{1/2}} exp\{-\frac{1}{2}(X - \mu_i)^T \Sigma_i^{-1}(X - \mu_i)\} \tag{4.1}$$

- For Bernoulli features, each component i is assumed to be a distribution in the BMM:

$$p(Y|\theta_{ber,i}) = \prod_{i=1}^{B} \alpha_i^Y (1 - \alpha_i)^{(1-Y)} \tag{4.2}$$

4.2.1. How to apply EM to GBMM

Firstly it is needed to find the distribution function that includes all the observed data points given the latent variable that explains the correlations among them, Z. Since it is assumed that the data coming from the two different distributions are independent:

$$f(O|Z,\theta) = f(X|Z,\theta_{gauss})f(Y|Z,\theta_{ber})f(Z|\theta) \tag{4.3}$$

Then, the complete *loglikelihood* function can be written as:

$$lnf(O|Z,\theta) = ln[f(X|Z,\theta_{gauss})f(Y|Z,\theta_{ber})f(Z|\theta)] \tag{4.4}$$

In the E step of the EM algorithm, the expectation of 4.4 is computed:

$$\begin{aligned}
&Q(\theta,\theta^{old}) \\
&= \mathbb{E}_Z(lnf(O|Z,\theta) \\
&= \mathbb{E}_Z[ln(f(X|Z,\theta_{gauss})f(Y|Z,\theta_{ber})f(Z|\theta))] \\
&= \mathbb{E}_Z\left[ln\left(\prod_{i=1}^{K} f(x_i|z_i,\theta_{gauss})f(y_i|z_i,\theta_{ber})f(z_i|\theta)\right)\right] \\
&= \mathbb{E}_Z\left[ln\left(\prod_{i=1}^{K}\left(\prod_{j=1}^{G} f(x_{i,j}|z_i,\theta_{gauss})\right)\left(\prod_{k=1}^{B} f(y_{i,k}|z_i,\theta_{ber})\right)f(z_i|\theta)\right)\right] \\
&= \sum_{i=1}^{K}\mathbb{E}_{z_i}\left[ln\left(\prod_{j=1}^{G} f(x_{i,j}|z_i,\theta_{gauss})\right) + ln\left(\prod_{k=1}^{B} f(y_{i,k}|z_i,\theta_{ber})\right) + lnf(z_i|\theta)\right] \\
&= \sum_{i=1}^{K}\mathbb{E}_{z_i}\left[\left(\sum_{j=1}^{G} lnf(x_{i,j}|z_i,\theta_{gauss})\right) + \left(\sum_{k=1}^{B} lnf(y_{i,k}|z_i,\theta_{ber})\right) + lnf(z_i|\theta)\right] \\
&= \sum_{i=1}^{K}\mathbb{E}_{z_i} lnf(x_i|z_i,\theta_{gauss}) + \sum_{i=1}^{K}\mathbb{E}_{z_i} lnf(y_i|z_i,\theta_{ber}) + \sum_{i=1}^{K}\mathbb{E}_{z_i} lnf(z_i|\pi) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})lnf(x_n|\theta_{gauss,k}) + \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})lnf(y_n|\theta_{ber,k}) + \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})lnf(z_n|\pi_k) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})ln(\pi_k f(x_n|\theta_{gauss,k})f(y_n|\theta_{ber,k}))
\end{aligned} \tag{4.5}$$

Where, following the same reasoning done for GMM and BMM:

$$\begin{aligned}
\gamma(z_{n,k}) &= p(z_{n,k}=1|O) = E[z_{n,k}] \\
&= \frac{\pi_k f(x_n|\theta_{gauss})f(y_n|\theta_{ber})}{\Sigma_{j=1}^{K}\pi_j f(x_j|\theta_{gauss})f(y_j|\theta_{ber})}
\end{aligned} \tag{4.6}$$

And so, the complete *log likelihood* takes a much simpler form to implement in code:

$$lnf(O|Z,\theta) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})ln(f(o_n|\theta_k)) \tag{4.7}$$

where

$$f(o_n|\theta_k) = \pi_k f(x_n|\theta_{gauss,k})f(y_n|\theta_{ber,k}) \tag{4.8}$$

being $N = B+G$ the total number of data points, K the number of clusters and o_n each of the data samples of the complete data set O.

4.2.2. Iterative solution for EM in GBMM

The above process can be summarized in these steps, similar to the ones defined for GMM and BMM, and the general one defined in section: 3.2.1.
1) Initialize the parameters θ: μ_k, Σ_k, α_k, π_k.
2) **E step**: evaluate $\gamma(z_{n,k})$ (4.6) using the current parameter values.
3) **M step**: update the parameters:

$$\alpha_k^{new} = \overline{y}_k$$

$$\mu_k^{new} = \frac{1}{N_k}\Sigma_{n=1}^{N}\gamma(z_{n,k})x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k}\gamma(z_{n,k})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

where $\overline{y}_k = \frac{1}{N_k}\Sigma_{n=1}^{N}\gamma(z_{n,k})y_n$ and $N_k = \Sigma_{n=1}^{N}\gamma(z_{n,k})$.

4) Evaluate the log likelihood (4.7) to check for convergence, if not, return to *step 2*.

4.3. Python Implementation

Implemenation using open source libraries like Numpy [14], Scipy [15], Pandas [16]. The code can be found in a personal GitHub repository developed during this thesis: github.com/franciscocobo/GaussianBernoulliMixtureModel.

5. MODEL VALIDATION

In this chapter, several toy examples with ground truth will be developed in order to test that the implementation of the mixture models was done correctly.

5.1. Introduction

Different tests for validating the defined model, as well as their implementation, are proposed in this chapter; toy examples with ground truth are developed. Later, for the real data (Gaussian), several techniques are applied in order to determine interesting metrics like feature selection, scatter separability and BIC (Bayesian Information Criterion).

5.2. Toy example with ground truth for testing the model

Having in mind that mixture models assume that a hidden variable explains the correlation of the data and thus their separability in *clusters*, it is possible to pre-define a data set that is governed by that variable and later try to get the grouping of the data having no access to the *latent variable*. That is what is done in the toy examples proposed for the 3 mixture models studied: GMM, BMM and GBMM.

5.2.1. GMM

The first test for the Python GMM implementation was done in order to check if the model could reconstruct the parameters for a bi-dimensional Gaussian distribution. 300 samples were generated and that was the information that the algorithm worked with. The parameters for the distribution were: $\mu = ([0, 5], [1, 0])$ $\Sigma = ([[2, 0], [0, 1.5]], [[3, 0], [0, 3]])$ and $\pi = ([0.4, 0.6])$ and the reconstructed values were the following:

0,899	0,022
0,183	4,807

(a) μ reconstructed

0,592
0,411

(b) π reconstructed

1,928	0,000
0,000	1,700

(c) Σ_1 reconstructed

3,331	0,000
0,000	2,474

(d) Σ_2 reconstructed

Table 5.1. Results of first toy example GMM

These results plotted on top of the data points look as follows:

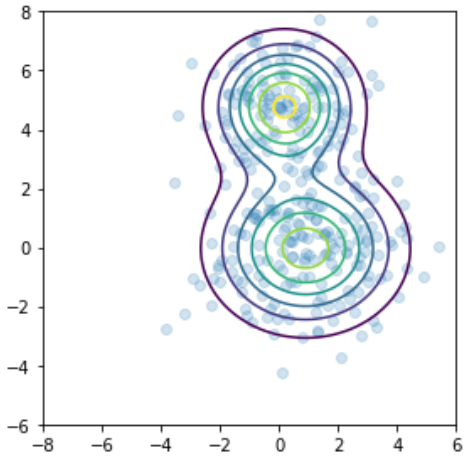


Fig. 5.1. Plot of first toy example in the GMM

Also, having the covariance matrices and the mean vector, it would be possible to *generate* new samples of data. One of the conditions of the EM algorithm is that it always outputs a result that improves the *log likelihood* of the observation, by definition. In the following plot the *-log likelihood* is plotted against the number of iterations of the EM algorithm for GMM.

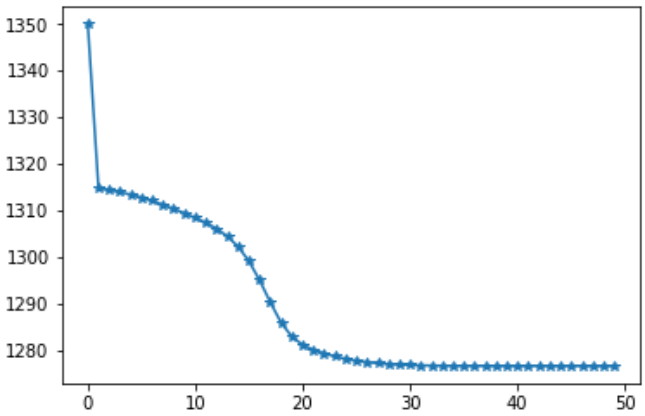


Fig. 5.2. *-log likelihood* against the number of iterations of the EM algorithm

It can be easily seen that in the first iterations of the algorithm, the improvement of the model is much bigger than in later stages. From the iteration number 25, it starts to converge. This test that was just done is one way to check if the model could reconstruct well the parameters that generated the data taken as input. However, as mentioned many times before, it is assumed that a hidden variable explains correlations among the data. In order to test the reconstruction of that *latent variable* into the $\gamma(z_{n,k})$, data coming from a known distribution governed by that variable was generated. In this example (figure 5.2), 3 clusters of data were defined. The variable Z was defined so that 50 % of the data came from cluster 0, 25 % from cluster 1 and 25 % from cluster 2, where $\mu_0 = ([0,0])$, $\mu_1 = ([5,5])$, $\mu_2 = ([2,2])$, $\Sigma_0 = ([[2,0],[0,0.1]])$, $\Sigma_1 = ([[0.1,0],[0,2]])$, $\Sigma_2 =$

([[2, −3], [−1, 2]]). In the following figure it can see the representation of the data points. It is clear that cluster 1 is more differentiated from the other two. Also, there are some data points that are halfway between two clusters, like data point 11 (between cluster 0 and 2). Al of this will be reflected in the results of the EM algorithm.

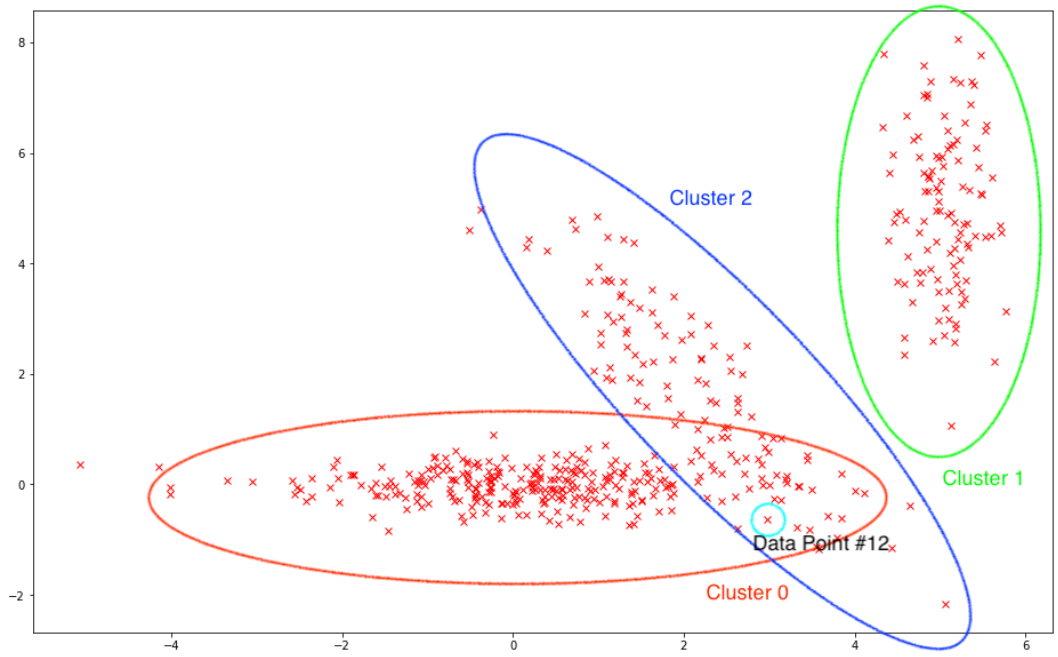


Fig. 5.3. Data generated for the second toy example GMM

In Table 5.2, the hidden variable Z and its reconstruction $\gamma(z_{n,k})$ are represented for the first 30 samples of the data.

Note that the order of the clusters indexes does not need to match the numbering in the Z. In this example, the first column corresponds to cluster 1, the second to cluster 2 and the third to cluster 0. Colours are used for better illustration. $\gamma(z_{n,k})$ represents the probabilities of coming from each cluster of each data point. There are some data points that have lower probabilities of belonging to an specific cluster, and it can checked that those data points correspond to those with features similar to the means of more than one of the clusters, like data point 12 (marked in the plot: [2,627, -0,801]). This kind of information is just an example of the use of the generative models, as it can be extracted many conclusions from the data. It does not only group the data in clusters, but also lets see how *hard* or *strong* is the decision of a data point belonging to an specific cluster, and also how likely is to group that data point into another cluster. From $\gamma(z_{n,k})$ it can also be extracted the conclusion that cluster 1 has always a *harder* decision that clusters 0 and 2, as it was previously guessed from the distribution of the data.

1
2
0
1
1
2
2
1
0
2
0
1
2
0
0
2
2
0
0
0
2
2
2
1
0
0
0
2
2
2

(a) Z

Data point #:			
0	1	0	0
1	0	1	0
2	0	0,012	0,988
3	1	0	0
4	1	0	0
5	0	1	0
6	0	1	0
7	1	0	0
8	0	0,008	0,992
9	0	0,101	0,899
10	0	0	1
11	1	0	0
12	0	0,529	0,471
13	0	0,005	0,995
14	0	0,075	0,925
15	0	1	0
16	0	0,868	0,132
17	0	0,05	0,95
18	0	0,078	0,922
19	0	0,008	0,992
20	0	0,167	0,833
21	0	1	0
22	0	0,913	0,087
23	1	0	0
24	0	0,118	0,882
25	0	0,01	0,99
26	0	0,004	0,996
27	0	0,998	0,002
28	0	1	0
29	0	1	0

(b) $\gamma(z_{n,k})$

Table 5.2. Results of second toy example in the GMM

5.2.2. BMM

In the case of binary data, the problem is much less visual, as binary data cannot be represented as easily as real (considered as gaussian) data. However, using the parameters of the EM model it is easy to check how well it is performing. Again, 3 clusters of data were defined, each with 1/3 probability of appearance. 300 data points were generated given the probability of activation of each feature, being those 10:
$cluster0 : \alpha_0 = [0.1, 0.9, 0.1, 0.9, 0.1, 0.9, 0.1, 0.9, 0.1, 0.9]$
$cluster1 : \alpha_1 = [0.1, 0.1, 0.1, 0.1, 0.1, 0.9, 0.9, 0.9, 0.9, 0.9]$
$cluster2 : \alpha_2 = [0.9, 0.9, 0.9, 0.9, 0.9, 0.1, 0.1, 0.1, 0.1, 0.1]$
The parameter α, which reconstructs the probabilities of activation of the features from the data points that belongs each specific cluster had the following form:

CLUSTER 1	0,142	0,077	0,138	0,081	0,094	0,869	0,949	0,897	0,928	0,897
CLUSTER 2	0,845	0,907	0,924	0,924	0,958	0,135	0,104	0,086	0,104	0,117
CLUSTER 0	0,126	0,921	0,064	0,939	0,097	0,844	0,111	0,880	0,093	0,910

Table 5.3. α reconstructed toy example in the BMM

The model reconstructs correctly the probabilities of activation.

As we can see in Table 5.4, the model reconstructs the *latent variable* without any problems. It can be seen that the decision between cluster 2 and cluster 1 is never questioned, that is because their features are clearly differentiated. However, the decision between cluster 0 and any of the other two clusters is not as *hard* due to a higher similarity in the probability of activation of the features, as it can be seen in data point 22 (marked in blue). This data point had the following features: [0, 0, 0, 1, 0, 1, 1, 1, 0, 1], which is in fact somehow between the characteristics (probabilities of activation: 5.3) of clusters 1 and 0.

The variation of the *-log likelihood* is as follows:

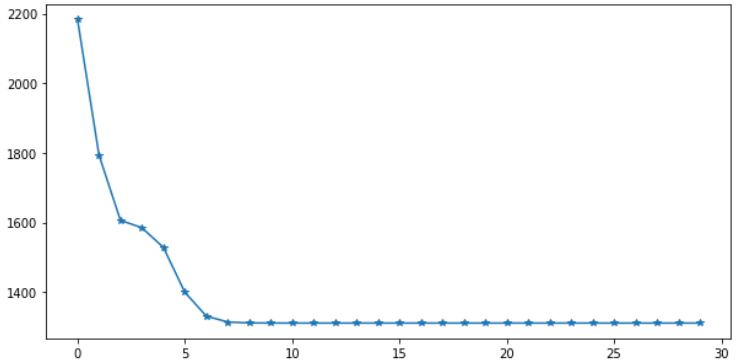


Fig. 5.4. *-log likelihood* against the number of iterations in the BMM algorithm

where it can be seen that after the iteration number 7 or so, the algorithm converges.

2
1
1
2
0
2
1
1
0
1
1
1
2
1
1
1
0
0
0
0
1
0
1
1
1
2
1
2
1
0

(a) Z

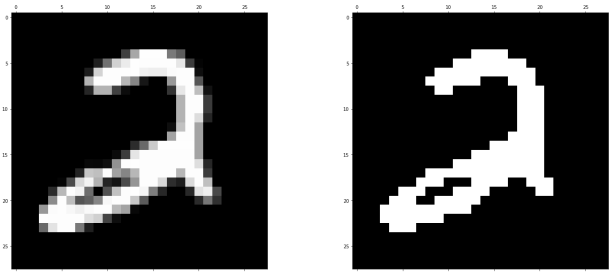
Data point #:			
0	0	1	0
1	1	0	0
2	1	0	0
3	0	1	0
4	0	0	1
5	0	1	0
6	1	0	0
7	1	0	0
8	0	0	1
9	1	0	0
10	0,992	0	0,008
11	0,992	0	0,008
12	0	1	0
13	0,992	0	0,008
14	1	0	0
15	1	0	0
16	0	0	1
17	0,013	0	0,987
18	0,006	0	0,994
19	0	0	1
20	0,996	0	0,004
21	0,005	0	0,995
22	0,402	0	0,598
23	0,988	0	0,012
24	1	0	0
25	0	1	0
26	1	0	0
27	0	1	0
28	1	0	0
29	0	0	1

(b) $\gamma(z_{n,k})$

Table 5.4. Results of toy example in the BMM

In order to understand better the purpose and conclusions of the EM algorithm applied to BMM, the following test was done: taken the binarized database of images of handwritten digits from NMIST [19], the data was given as input to the defined BMM, as well as the known number of clusters (10, numbers 0 to 9). The results of the α for each cluster were plotted, using a white-gray-back scale depending on the value of the parameters (from 0 to 1), which, being re-scaled to a 28x28 matrix, outputs a reconstruction of the written number of each of the clusters defined. Using this test, it can be seen in a very visual way how the algorithm performs.

The NMIST database gives a set of handwritten numbers in a gray scale, which was binarized as it can be seen in Figure 5.5, in order to give as input to the BMM algorithm a matrix of binary values.



(a) Gray scale (b) Binarized: b&w

Fig. 5.5. Handwritten example of number 2 in the NMIST [19] database

The complexity of this example is much higher than the one of the previous case, because of the number of data points, being in this case a matrix of 70000x784. Even though in the following graph the algorithm seems to converge soon, the improvements are seen up to more than 700 iterations.

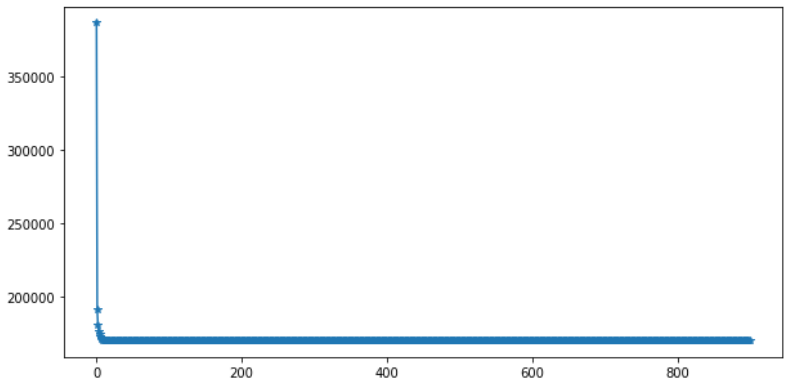
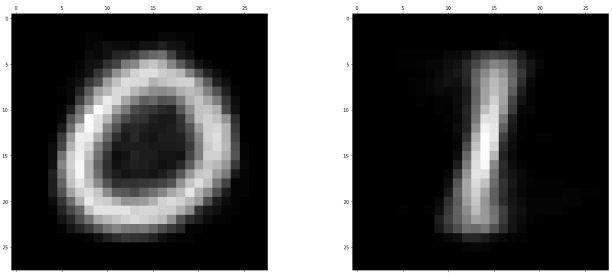


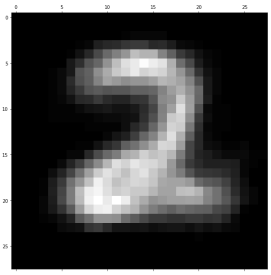
Fig. 5.6. *-log likelihood* against the number of iterations in the BMM algorithm with NMIST data

Plotting the α parameter (figure: 5.7), being re-scaled to 28x28 matrices, which outputs the probability of activation of each of the features in each cluster outputs the following plotted in a gray scale. It can be seen that the clusters means represent a mixture of all the hand written numbers that correspond to the same actual number.

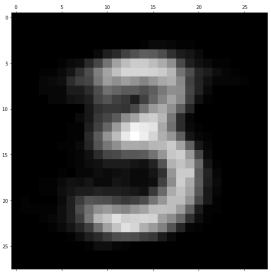


(a) 0 reconstructed

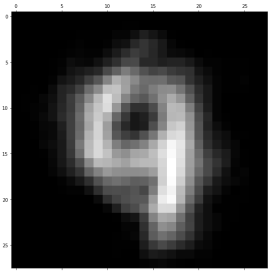
(b) 1 reconstructed



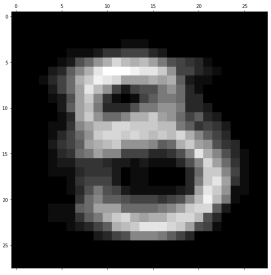
(c) 2 reconstructed



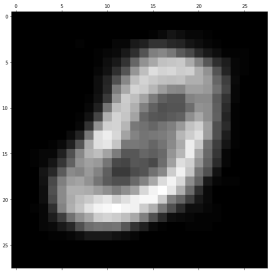
(d) 3 reconstructed



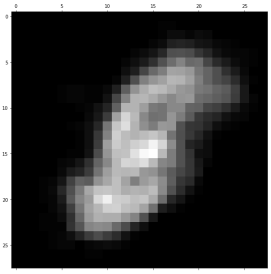
(e) 4 reconstructed



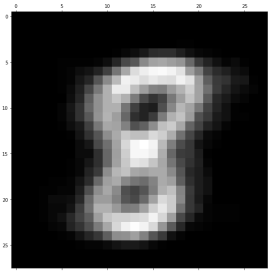
(f) 5 reconstructed



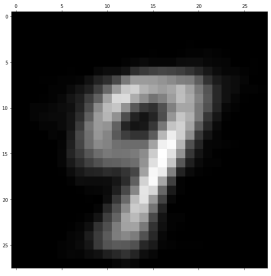
(g) 6 reconstructed



(h) 7 reconstructed



(i) 8 reconstructed



(j) 9 reconstructed

Fig. 5.7. Cluster reconstruction in the BMM with NMIST data

5.2.3. GBMM

Finally, the last toy example with ground truth proposed for validation of the model is a GBMM test. Data coming from the same Z were generated, having for each cluster a particular μ_k and Σ_k for the real data, as well as a α_k for the binary data. 3 clusters of data were again defined (50% of the data coming from cluster 0 and 25% from the other two clusters), with the following parameters:

$cluster0$:
$\alpha_0 = [0.5, 0.9, 0.1, 0.9, 0.1, 0.9, 0.1, 0.9, 0.1, 0.9]$
$\mu_0 = [0, 0]$
$\Sigma_0 = [[0.1, 0], [0, 2]]$
$cluster1$:
$\alpha_1 = [0.1, 0.1, 0.1, 0.1, 0.1, 0.9, 0.9, 0.9, 0.9, 0.9]$
$\mu_1 = [5, 5]$
$\Sigma_1 = [[2, 0], [0, 0.1]]$
$cluster2$:
$\alpha_2 = [0.9, 0.9, 0.9, 0.9, 0.9, 0.1, 0.1, 0.1, 0.1, 0.1]$
$\mu_2 = [2, 2]$
$\Sigma_2 = [[3, -2], [-2, 2]]$

The reconstructed parameters were the following:

CLUSTER 2	0,882	0,911	0,970	0,971	0,970	0,030	0,147	0,206	0,147	0,059
CLUSTER 0	0,457	0,971	0,114	0,971	0,143	0,943	0,171	0,971	0,086	0,943
CLUSTER 1	0,214	0,179	0,179	0,142	0,214	0,999	0,964	0,999	0,893	0,999

Table 5.5. α reconstructed toy example in the GBMM

CLUSTER 2	1,958	2,166
CLUSTER 0	-0,099	0,046
CLUSTER 1	5,258	4,805

Table 5.6. μ reconstructed in toy example in the GBMM

0,115	0,159
0,159	2,648

(a) Σ_0

2,096	-0,031
-0,031	0,118

(b) Σ_1

3,174	-2,138
-2,138	1,606

(c) Σ_2

Table 5.7. Σ reconstructed in toy example in the GBMM

It is interesting to notice that the probabilities of activation of the $\gamma(z_{n,k})$ (figure 5.8) are 1 (after rounding them up) because of having more information than in the previous examples, the model can make a *harder* decision.

2
0
1
2
2
1
0
0
1
2
0
2
2
2
2
2
2
1
0
2
0
2
2
0
0
0
1
2
1
2

(a) Z

1	0	0
0	1	0
0	0	1
1	0	0
1	0	0
0	0	1
0	1	0
0	1	0
0	0	1
1	0	0
0	1	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
0	0	1
0	1	0
1	0	0
0	1	0
1	0	0
1	0	0
0	1	0
0	1	0
0	1	0
0	0	1
1	0	0
0	0	1
1	0	0

(b) $\gamma(z_{n,k})$

Table 5.8. Results of toy example GBMM

After these 3 test were done, it can be concluded that the model proposed works well.

5.3. Interesting metrics

One of the interesting properties of Generative Modeling is that it allows to extract lot of information from the data. Because of having followed a pure statistical approach, the clustering algorithm not only outputs the separation of the data points in groups but also

many parameters that can be interpreted and studied. Several metrics will be defined in order to quantify how important are the different features. The conclusions obtained in this section will be later applied for the analysis of the patients data. The metrics proposed are only related to real data, considered as Gaussian, and so only the GMM toy example will be taken into consideration in this section.

5.3.1. Feature selection and scatter separability

It is particularly important in Generative Modeling to select the right variables to study, the features of the data points that are more significant and give the most information. In the particular case of this thesis, this information is partly given by the doctors, who generated the database and are experts on the field. However, in many other cases there will not be available the help of an external expert figure who says which variables are and are not important. In any case, the study of the variation of the features within a cluster and in relation with other clusters, is of great importance as it will give much information about how the clustering was performed and how accurate its result is.

The *separability* of a cluster measures how well it can be identified with respect to other clusters. This will be reflected in the $\gamma(z_{n,k})$ result of the algorithm, as the higher the probabilities of belonging to one cluster or another means that the result is more reliable. But it can also be studied from the already mentioned feature selection metrics.

As proposed in [20], several metrics are introduced that help in finding the features that are most significant within a cluster of data:

S_w measures the variation of the features within the clusters. The higher the value, the more variation of the particular feature inside data coming from a cluster. Therefore it means that the higher the variation, the less homogeneous the clusters are according to that particular feature.

$$S_w = \sum_{j=1}^{k} \pi_j \, \mathbb{E}[(X - \mu_j)(X - \mu_j)^T | \omega_j] = \sum_{j=1}^{k} \pi_j \Sigma_j \tag{5.1}$$

S_b measures the variation of the features of the data points in relation to other data points in other clusters, so the higher the values in the result matrix, the better the clusters will be differentiated from other clusters.

$$S_b = \sum_{j=1}^{k} \pi_j (X - M_o)(X - M_o)^T \tag{5.2}$$

Where:

$$M_o = \mathbb{E}[X] = \sum_{j=1}^{k} \pi_j \mu_j \tag{5.3}$$

Being the mean of the features inside the whole dataset.

Several examples will be provided. Firstly, these metrics are applied to the data generated for the second toy example of the GMM.

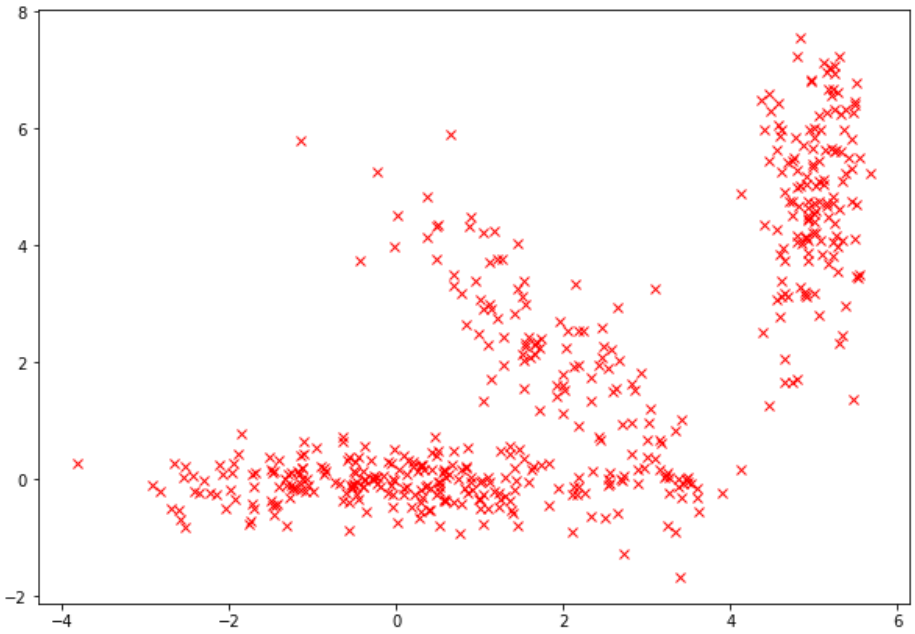


Fig. 5.8. Data generated for the second toy example in the GMM

x axis	y axis
2,003	1,770

Table 5.9. M_o toy example GMM

	x axis	y axis
x axis	3,977	4,092
y axis	4,092	4,213

Table 5.10. S_b toy example GMM

	x axis	y axis
x axis	1,342	0,435
y axis	0,435	1,322

Table 5.11. S_w toy example GMM

It can be checked that the variation within a cluster is much less that the variation in relation with other clusters, which means that both features studied (axis x and axis y) give relevant information to the model. In this example the metrics are not completely clear so in the next examples more extreme cases will be discussed.

In this second example the dataset is forced to have clusters with a lot of variation in the x axis, but less differentiation within the y axis, what can be checked that is reflected in the scatter separability metrics introduced.

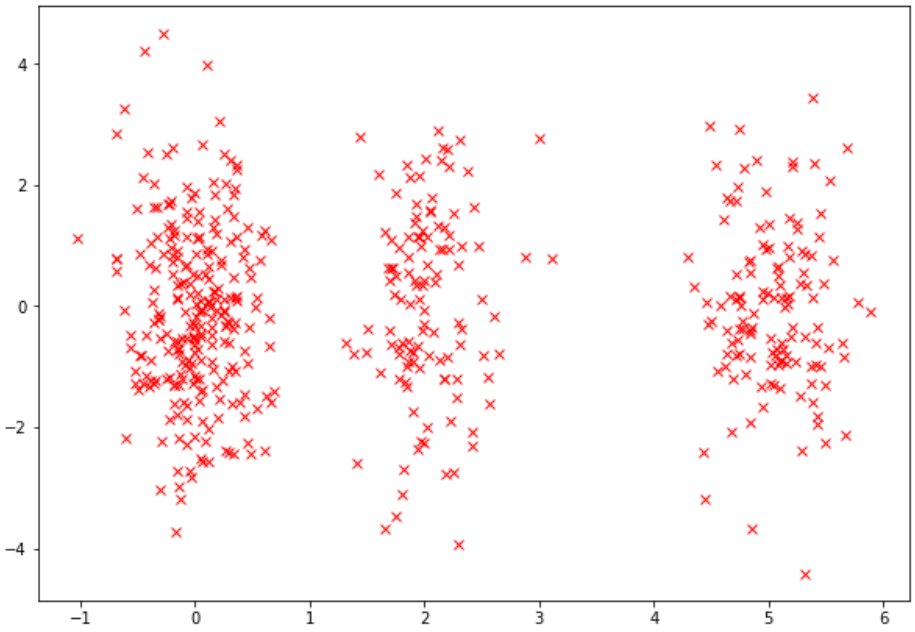


Fig. 5.9. Data generated for scatter separability test 1

x axis	y axis
1,792	-0,048

Table 5.12. M_o second dataset GMM

	x axis	y axis
x axis	4,020	0,009
y axis	0,009	0,004

Table 5.13. S_b second dataset GMM

The differences among the clusters are just in the x axis.

	x axis	y axis
x axis	0,460	0,046
y axis	0,046	2,077

Table 5.14. S_w second dataset GMM

Within a particular cluster, the variation is basically in the y axis.

In the last example, the clusters vary much more in the y axis than in x axis.

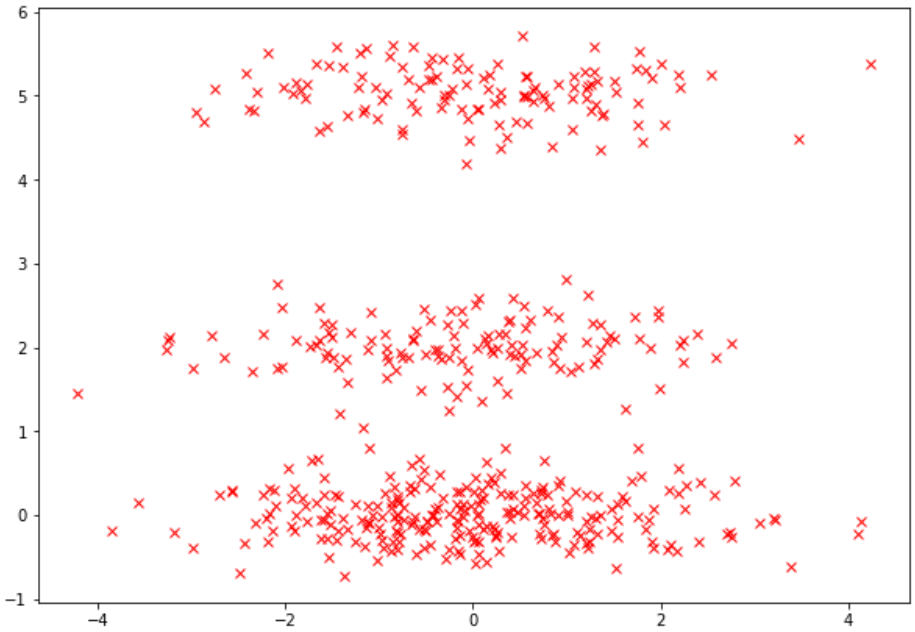


Fig. 5.10. Data generated for scatter separability test 2

x axis	y axis
0,014	1,853

Table 5.15. M_o third dataset GMM

	x axis	y axis
x axis	0,044	0,037
y axis	0,037	4,461

Table 5.16. S_b third dataset GMM

The differences among the clusters are just in the y axis.

	x axis	y axis
x axis	2,264	0,018
y axis	0,018	0,097

Table 5.17. S_w third dataset GMM

Within a particular cluster, the variation is basically in the x axis.

5.3.2. Bayesian Information Criterion

One of the characteristics of a mixture model is that it assumes that the data is coming from several groups that are inferred. However, it not only unknown the variable that groups each data point, but also the number of groups, or *clusters* that exist in the database. That is the reason for the Bayesian Information Criterion (BIC) to be introduced, as it is a metric than combines the higher *log likelihood* that will imply an increment in the number of guessed clusters (being the maximum when the number of clusters equals the number of data points, which would not make sense) and the increase in the complexity of the model (increasing for a higher number of clusters) [21].

The BIC aims to pick the model with the largest possible marginal likelihood:

$$K^* = \arg\max_K p(\mathbf{X}|K) = \arg\max_K \int_\theta p(\mathbf{X}|\theta, K)p(\theta|K)d\theta \tag{5.4}$$

However, it is general difficult to compute the mentioned marginal likelihood. If it is assumed that the prior distribution $p(\theta|K)$ is a gaussian and it is very broad, the marginal likelihood can be approximated as:

$$-\log p(\mathbf{X}|K) \approx \frac{K \log N}{2} - p(\mathbf{X}|\theta_{MAP}, K) \tag{5.5}$$

Where the first term makes the likelihood increase with the number of clusters and the second term is a penalty term that also increases with the number of clusters. Applying the BIC to the second toy example provided for the GMM, the following result is obtained:

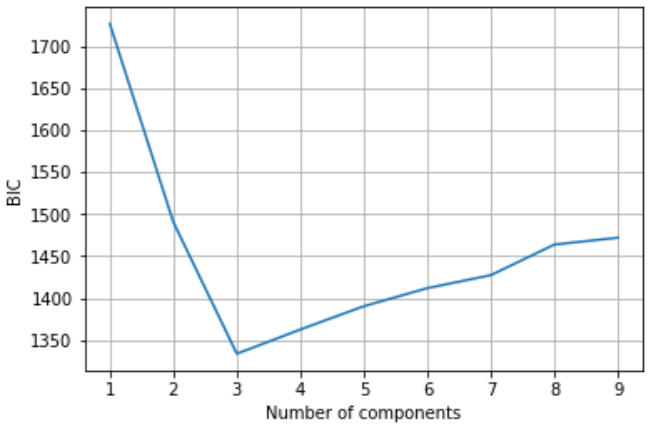


Fig. 5.11. BIC toy example GMM

It can be seen that the optimum number of clusters obtained matches the number of predefined ones, 3 clusters.

6. EXECUTING THE MODEL WITH REAL DATA

In this chapter, the models proposed in previouses chapters will be executed with the database used for the thesis.

6.1. Dataset

The dataset used for this thesis was a medical database that could be accessed because of an university collaboration with Madrid's Hospital Gregorio Marañón. It cointained information from 134 patients with Acute Myeloid Leukemia. The data availabe was both real data and binary data that was analyzed as Gaussian data and Bernoulli/Categorical data respectively. Columns of the data set with more that 20-30% NaNs (missing data) were discarded and the rest of the NaNs were completed with the average of the column in the case of real data and the most common value in the case of binary data. This approximation could be done better in a more complex way, but it worked for this problem because the amount of missing data was either enormous (discarded) or very small.

6.2. About the analysis

As mentioned before, in the particular analysis carried out in this thesis, the expertise of the doctors helped in the way that is was known the most probable number of clusters that existed in the database (3) and also which features should be taken into consideration. The data from 134 patients were used in the analysis, and the used features were: age, leukocytes, platelets, lactate dehydrogenase (LDH), percentage of blasts (treated as gaussian data) and Eastern Cooperative Oncology Group (ECOG) scale, AML type (*de novo* and secondary), gender, Methyl Orange (MO) indicator, NPML1 and FLT3 mutations, and cytogenic risk (treated as binary-categorical bernoulli data). Also the global survival of the patients of each cluster was included in the results that were sent to the doctors, but it was not used in the analysis as it was not a diagnostic-time variable.

	% OF MISSING DATA	DATA TREATED AS
Age	0,00%	Gaussian
Leukocytes	0,70%	Gaussian
Platelets	0,70%	Gaussian
LDH	2,90%	Gaussian
Percentage of blasts	4,50%	Gaussian
ECOG scale	8,90%	Binary
AML type	0,00%	Binary
Gender	0,00%	Binary
MO Indicator	0,00%	Binary
NPM1/FLT3	0,00%	Categorical
Cytogenic Risk	0,00%	Categorical

Table 6.1. Data used for the analysis

6.3. Results

In the following sections the result of the analysis of the gaussian data, the binary and categorical data, and then the analysis with all the data together will be presented. Only a part of the analysis will be shown due to the privacy of the data used, however, an extended technical report was sent to the doctors experts on the field that drove to the conclusions that will be later mentioned. Data was normalized between 0 and 1.

6.3.1. Gaussian Data (GMM)

Firstly the *-log likelihood*: $-ln\{p(X|\pi,\mu,\Sigma)\} = -\Sigma_{n=1}^{N} ln\{\Sigma_{k=1}^{K}\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\}$ (equation 3.20) is plotted against the number of iterations of the algorithm, where it can be seen that the GMM algorithm converges after iteration number 25.

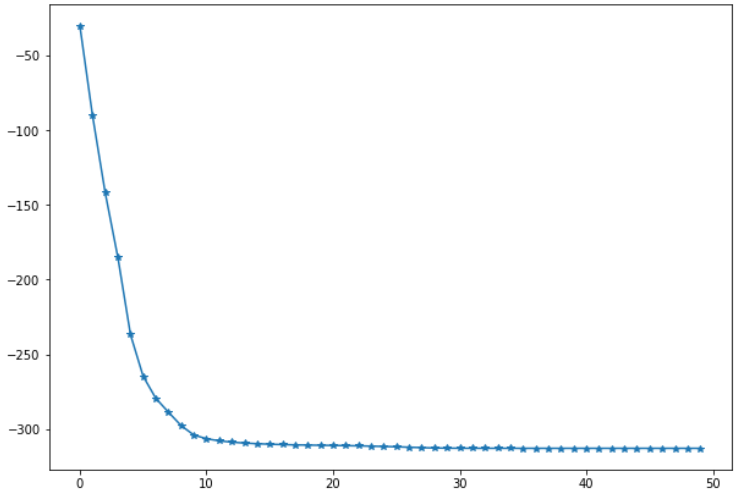


Fig. 6.1. *-log likelihood* for GMM with patients data

Secondly, the before-mentioned BIC criterion is applied to the dataset and the following result is obtained:

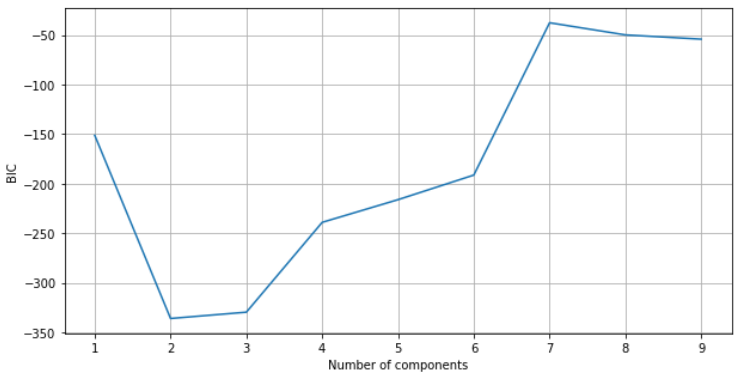


Fig. 6.2. BIC for GMM with patients data

It can be seen than the database could be divided in either 2 or 3 clusters, as the doctors divide the patients in 3 groups, it was chosen that number as the number of predefined clusters that the algorithm had to work with. However, the following results explain how the division of the patients in 3 clusters is not so clear.

In the following tables, the scatter separability metrics S_b (variation of the variables between clusters - equation 5.2) and S_w (variation of the variables within a cluster - equation 5.1) are shown.

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts
Age	0,004	0,007	0,002	0,005	0,008
Leukocytes	0,007	0,011	0,003	0,008	0,014
Platelets	0,002	0,003	0,002	0,002	0,004
LDH	0,005	0,008	0,002	0,006	0,009
Percentage of Blasts	0,008	0,014	0,004	0,009	0,017

Table 6.2. S_b of the patients data in the GMM

From S_b it can be concluded that the variables that define the most a particular cluster (the ones that differentiate them from the others) are the leukocytes, the LDH and the percentage of blasts. On the other hand, the age and the platelets do not vary much from one cluster to another, they are not key variables in the separation of the patients in each cluster.

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts
Age	0,054	0,004	0,004	0,004	0,005
Leukocytes	0,004	0,024	0,003	0,008	0,010
Platelets	0,004	0,003	0,043	0,005	0,004
LDH	0,004	0,008	0,005	0,029	0,005
Percentage of Blasts	0,005	0,010	0,004	0,005	0,057

Table 6.3. S_w of the patients data in the GMM

From S_w it can be concluded that the variables that vary the most within a particular cluster are the age, the platelets (these two make sense as they are not essential in the definition of the clusters) and the percentage of blasts. Also, the leukocytes and the platelets vary within each cluster, this is because the division in the 3 clusters is not very clear.

Finally, in the following table the mean values (unnormalized) for each of the characteristics of the patients in every cluster is provided, as well as the global mean of the whole database for comparison.

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts	Survival time (days)
CLUSTER 1	50	127251	79800	1154	78	921
CLUSTER 2	54	22851	57718	600	59	862
CLUSTER 3	55	5363	123033	256	49	920
TOTAL MEAN	55	41170	87826	620	61	896

Table 6.4. Results for the GMM on the patients data

The previous conclusions obtained from S_b and S_w apply also here as there is not a big difference neither in age nor in platelets in the different clusters, but there is in leukocytes,

LDH and percentage of blasts. It can be also checked that the survival time of the patients (variable not used in the analysis) is very similar in the 3 groups. Therefore it can be concluded that the differences among the three clusters defined are not so cleared and it is probably necessary to include the binary variables to obtain more significant results.

6.3.2. Binary Data (BMM)

Again, firstly the *-log likelihood*:
$lnp(X, Z|\alpha, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{n,k} \left(ln\pi_k + ln \sum_{i=1}^{D} x_{n,i} ln\alpha_{k,i} + (1 - x_{k,i}) ln(1 - \alpha_{k,i}) \right)$ (equation 3.41) is plotted against the number of iterations of the algorithm, where it can be seen that the BMM algorithm converges after iteration number 15.

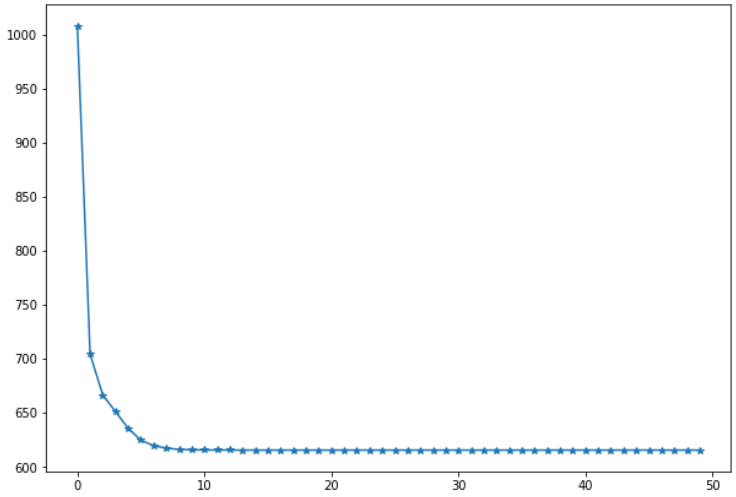


Fig. 6.3. *-log likelihood* for BMM with patients data

In the following table, the probability of activation of the different binary variables studied (and the categorical ones, studied using *one-hot encoding* are shown.

	ECOG	LMA	Gender	MO Indicator	NPM1-/FLT3-	NPM1+/FLT3-	NPM1+/FLT3+	NPM1-/FLT3+	Favorable Cyt. Risk	Intermediate Cyt. Risk	Unfavorable Cyt. Risk	Survival time (days)
CLUSTER 1	0,438	0,913	0,442	0,737	0,035	0,512	0,328	0,173	0,024	0,965	0,034	1.022
CLUSTER 2	0,634	0,730	0,366	0,692	0,972	0,023	0,024	0,026	0,101	0,896	0,025	1.204
CLUSTER 3	0,696	0,574	0,493	0,304	0,897	0,037	0,025	0,091	0,028	0,025	0,971	433
TOTAL MEAN	0,550	0,683	0,363	0,501	0,634	0,125	0,009	0,021	0,015	0,570	0,299	896

Table 6.5. Results for the BMM on the patients data

Legend:
LMA type - 0: secondary (worst kind) 1: de novo
Gender - 0: Man 1: Woman

It can be seen a more clear division of the patients in 3 groups, each of them with different probabilities of activation for each of the variables under study. The results show two clusters with similar survival time of the patients, with very similar characteristics except for the genetic mutation NPM1/FLT3. The patients of Cluster 3 have much less

survival time and their cytogenic risk is mostly unfavorable. There is also in the third cluster a higher number of patients with the secondary type of LMA.

6.3.3. All Data (GBMM)

Again, the first plot shows the *-log likelihood*: $lnf(O|Z,\theta) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})ln(\pi_k f(o_n|\theta_k))$ (equation 4.7) is plotted against the number of iterations of the GBMM algorithm. The algorithm converges after iteration 40.

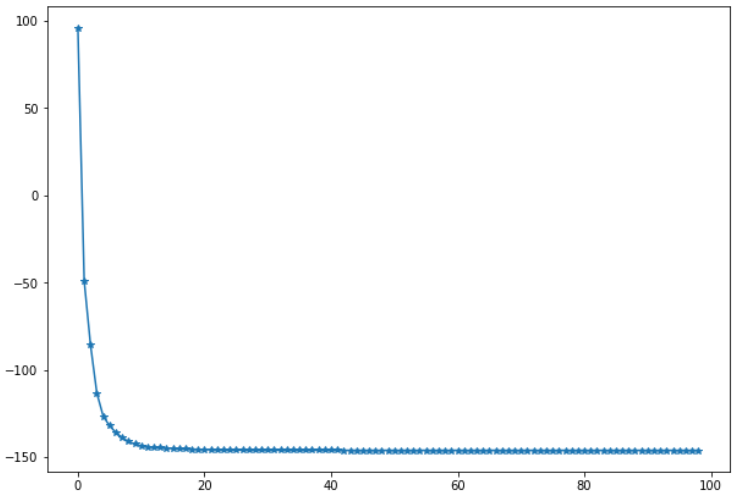


Fig. 6.4. *-log likelihood* for GBMM with patients data

In the case of the GBMM algorithm, the BIC criterion will have the same result of the GMM case as it takes only into consideration the Gaussian variables (in the particular way it was defined in this thesis).

However, as the S_b and S_w measure the variability of the Gaussian variables among the different clusters and within each of the clusters, and the clusters defined will be different in this case (GBMM), their results will be different.

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts
Age	0,005	0,009	0,003	0,005	0,010
Leukocytes	0,009	0,020	0,005	0,008	0,021
Platelets	0,003	0,005	0,002	0,003	0,006
LDH	0,005	0,008	0,003	0,005	0,009
Percentage of Blasts	0,010	0,021	0,006	0,009	0,023

Table 6.6. S_b of the patients data in the GBMM

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts
Age	0,053	0,006	0,002	0,004	0,002
Leukocytes	0,006	0,017	0,001	0,007	0,004
Platelets	0,002	0,001	0,042	0,004	0,002
LDH	0,004	0,007	0,004	0,029	0,005
Percentage of Blasts	0,002	0,004	0,002	0,005	0,052

Table 6.7. S_w of the patients data in the GBMM

The results obtained are very similar to the ones obtained for the GMM algorithm, and the key variables for the definition of the 3 clusters are the same (leukocytes, LDH and percentage of blasts). Also, the similar variability of all the variables within each of the particular clusters, leads to think that the division in 3 clusters might not be the best.

Finally, in Table 6.8 the mean of the Gaussian variables (unnormalized), and the probabilities of activation of the binary variables in each of the clusters and in the total dataset are shown. Clusters 1 and 2 have a similar survival time, better cytogenic risk and a higher number of patients with the *de novo* type of LMA. These clusters are differentiated by the NPM1/FLT3 mutation, the leukocytes, the LDH and the percentage of blasts. The patients in Cluster 3 have a much lower survival time: their cytogenic risk mostly unfavorable and more patients have the secondary type of LMA. Their number of leukocytes is specially low and the other Gaussian variables are not very different from the patients of the other clusters.

6.3.4. Clustering results

In Table 6.9 the $\gamma(z_{n,k})$ is shown for the 3 different algorithms for 37 random patients. Cluster 1 is represented in green, Cluster 2 in blue and Custer 3 in red. It can be seen how the patients jump from one cluster to another depending on which variables are taking into consideration in the analysis.

6.4. Conclusion

As a conclusion of the analysis, there are two main groups of patients. One of the groups involves patients with a survival time from the diagnosis of about 1050 days. The patients in this group have an intermediate cytogenic risk and most of them have the *de novo* type of LMA. Gender does not play a role in this division. This group could be divided in other two subgroups, having different NPM1/FLT3 mutation and different percentage of blasts, LDH and number of platelets. The other group contains patients with less survival time since the diagnosis, about 370 days. They are mostly determined by having an unfavorable cytogenic risk and the secondary type of LMA.

	Age	Leukocytes	Platelets	LDH	Percentage of Blasts	Survival time (days)	ECOG	LMA	Gender	MO Indicator	NPM1-/FLT3-	NPM1+/FLT3-	NPM1+/FLT3+	NPM1-/FLT3+	Favorable Cyt. Risk	Intermediate Cyt. Risk	Unfavorable Cyt. Risk
CLUSTER 1	48	110760	65715	1041	82	1070	0,483	1,000	0,446	0,765	0,232	0,219	0,366	0,211	0,057	0,830	0,140
CLUSTER 2	55	16155	95631	417	52	1130	0,606	0,757	0,420	0,693	0,695	0,265	0,015	0,043	0,034	0,984	0,001
CLUSTER 3	60	9919	96117	497	52	368	0,730	0,540	0,460	0,297	0,998	0,001	0,001	0,001	0,001	0,001	0,998
TOTAL MEAN	55	41170	87826	620	61	896	0,551	0,679	0,362	0,498	0,634	0,125	0,008	0,020	0,015	0,566	0,299

Table 6.8. Results for the GBMM on the patients data

0,002	0,998	0
1	0	0
0,927	0,073	0
0,077	0,923	0
1	0	0
1	0	0
1	0	0
1	0	0
0,002	0,998	0
0,986	0,014	0
0	1	0
0,047	0,953	0
0,158	0,842	0
1	0	0
0	0	1
0,998	0	0,001
1	0	0
1	0	0
0,599	0,401	0
0	1	0
1	0	0
0	0	1
1	0	0
0	1	0
0	1	0
0	1	0
0,001	0,999	0
0	1	0
1	0	0
0,041	0,959	0
0,805	0,195	0
0,001	0,999	0
0	0	1
1	0	0
0	0	1
0	1	0
0	0	1

(a) $\gamma(z_{n,k})$ GBMM

0	1	0
1	0	0
0,204	0,796	0
0,001	0,999	0
0,007	0,207	0,786
0,548	0,452	0
0,086	0,914	0
1	0	0
0,002	0,114	0,884
0,074	0,926	0
0	0	1
0,255	0,745	0
0,169	0,83	0,001
1	0	0
0,001	0,077	0,922
0,009	0,991	0
0,579	0,421	0
1	0	0
0,004	0,689	0,307
0	1	0
0,999	0,001	0
0	0,039	0,961
0,998	0,002	0
0	1	0
0	0,042	0,958
0	0,189	0,811
0	0,021	0,979
0	0	1
1	0	0
0,006	0,994	0
0,002	0,997	0,001
0	0,032	0,968
0	0,348	0,652
1	0	0
0	0,999	0,001
0	0,711	0,289

(b) $\gamma(z_{n,k})$ GMM

0,999	0,001	0
0,998	0,002	0
0,02	0,98	0
0,999	0,001	0
0,008	0	0,992
0,001	0,985	0,013
0,994	0,006	0
0,003	0	0,997
0,002	0,997	0
0,996	0,004	0
0,027	0,973	0
0,999	0,001	0
0,999	0,001	0
0	0,001	0,999
0	0,004	0,996
0,017	0	0,982
0,998	0,002	0
0,027	0,973	0
0,995	0,005	0
0,022	0,977	0,001
0,999	0,001	0
0	0,002	0,998
0,999	0,001	0
0,02	0,98	0
0,999	0,001	0
0,012	0,987	0
0,009	0,991	0
0,016	0,983	0,001
0,994	0,006	0
0,01	0,988	0,002
0,994	0,006	0
0,027	0,973	0
0	0,003	0,997
0,009	0,991	0
0	0,001	0,999
0,999	0,001	0

(c) $\gamma(z_{n,k})$ BMM

Table 6.9. Clustering results of all mixture models with the patients data

7. ANALYSIS OF THE EFFECT OF THE TRANSPLANT OF HEMOPOIETIC PROGENITORS

"In the second half of the XX century, the transplant of hemopoietic progenitors (TPH) ceased to be a desperate treatment with a high incidence of complications implying a high mortality, and became a curative treatment for thousands of patients with hematological neoplasias and other diseases" [22].

As requested by the doctors, the analysis of the effect of the TPH on the patients in the database was done and the conclusions of the analysis were the following.

The average survival time of the patients with AML since the diagnosis in the database was of 896 days. Those having a favorable cytogenic risk have a higher survival time than the average, 2882 days. The patients that have either an intermediate cytogenic risk or a unfavorable risk, are indicated for a TPH. The ones that did have a TPH have an average survival time of 1155 days. On the other hand, those with not a favorable cytogenic risk that did not have a TPH, have an average survival time of 604 days, proving the effectiveness of the transplant. Patients with a Methyl Orange (MO) indicator different than 1, regardless of their cytogenic risk, should also have the TPH. Those who did have the transplant have an average survival time of 685 days, in contrast to the 197 days of those with MO indicator different that 1 who did not have the TPH.

PATIENT GROUP	TPH	SURVIVAL TIME (in days)
Favorable cytogenic risk	No	2882
Intermediate or unfavorable cytogenic risk	Yes	1155
	No	604
MO indicator different than 1	Yes	685
	No	197

Table 7.1. Analysis of the effect of TPH

These results prove the effectiveness of the TPH.

8. OPINION OF THE DOCTORS ON THE RESULTS

The following are the opinions of the doctors on the results mentioned in Chapter 6.

About Cluster 1, the results make sense as the patients with LMA *de novo* are included in it and also those without unfavorable cytogenic risk.

Cluster 3 is also logic, it includes the patients with less survival time. They are patients with secondary LMA and unfavorable cytogenic risk.

However, Cluster 2 is a mix of patients of different characteristics that should not be included in the same group. This could be predicted as the BIC criterion and the scatter separability metrics suggested that the division of the patients in 3 clusters was not clear.

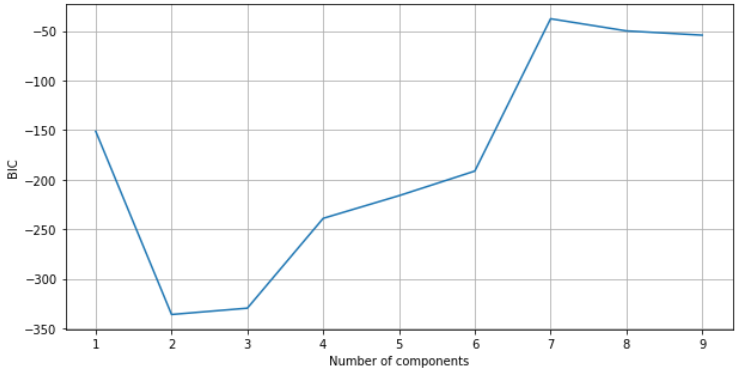


Fig. 8.1. BIC for GMM with patients data

Therefore, doctors validated the proposed results. Future work on the topic will try to find a better structure for the data by using hierarchical clustering models instead of non-overlapping clusters like in this thesis.

9. REGULATORY FRAMEWORK

In this thesis the regulatory framework is very important as the database used for the project contains data labeled as *special category* (very sensitive data, medical data). The brand new General Data Protection Regulation (GDPR) is an European Law that affects all countries members of the European Union and all companies and parties that operate in its territory. Its aim is to protect the privacy of all EU citizens as well as their data as the world tends to be more data-driven than ever. It is a very restrictive law that seeks for the user to be the real owner of its data, it obligates the companies to inform the customer how and why are they collecting the data and what they are used for. Their main changes with respect to previouses regulations are the following [23]:

- Increased territorial scope: even if the company is not from the EU, if their customers live inside the EU territory, they must apply the law.
- Penalties: the fines for breaking the data protection law have increased significantly.
- Consent: the conditions for the consent of the user to be valid have become much harder, and now the data policies of the companies must be easy to read and understand.

As the new regulation is so strong and hard, the scientific community in Spain was worried of how this would affect the academic studies. Particularly the bio-medical world, as for many studies sensitive data from patients were needed to be studied and analyzed [24]. Answering these claims, the Spanish Data Protection Agency *(Agencia Española de Protección de Datos)* realeased a clarification note assuring that bio-medical studies were not affected by the GDPR in Spain, as they are governed by Law 14/2007, 3th July *(Ley 14/2007, de 3 julio)*, where the rules for bio-medical studies are explained. *"La primera conclusión que puede extraerse de la literalidad de las normas que se han venido reproduciendo es la de que el Reglamento general de Protección de Datos no implica una alteración del marco normativo actualmente vigente en España en relación con el tratamiento de datos en el marco de la investigación biomédica.")* [25]. The GDPR specifies that in particular cases like the one here, the law of the states of the Union may prevail over the GDPR [26]. In the mentioned note, the guides of how to operate in studies with medical data are explained, they can be sum up in two main principles:

- The investigation must aim for the general good, and the data must be essential for the study.
- The patients must give their unequivocal and specific consent for the use of their data. Although it is considered also the variability of the scientific studies and how it not always possible to know at the time the data is being collected how will it be used. This is why it is considered a general patient consent, that states which kind of studies would they want their data to be used and which not. An example is given: In this way, to give an example, it would not be necessary, in order to guarantee the unequivocal and specific

nature of the consent, that it be provided for the realization of a specific investigation; not even for the realization of investigations in a very delimited branch, as for example, a certain type of cancer, but, taking into account the interpretation derived directly from the own regulation, will be sufficiently unequivocal and specific the consent given in relation with a branch extensive research, for example, oncological research, or even for more extensive areas. *("De este modo, por poner un ejemplo, no sería a preciso, para garantizar el carácter inequívoco y específico del consentimiento, que el mismo fuese prestado para la realización de una investigación concreta; ni siquiera para la realización de investigaciones en una rama muy delimitada, como por ejemplo, un determinado tipo de cáncer, sino que, teniendo en cuenta la interpretación derivada directamente del propio Reglamento, será suficientemente inequívoco y específico el consentimiento prestado en relación con una rama amplia de investigación, como por ejemplo, la investigación oncológica, o incluso para ámbitos mas extensos."*

Having in mind the importance and responsibility of working with sensitive data, special care was put while treating the data. The dataset was given completely anonymized, and it was not shared with anyone outside of the project. Also, both european and spanish mentioned regulations were followed. The Ethics Committee of the Madrid's Hospital Gregorio Marañón approved the project and the collaboration.

The main goal of the thesis was to help in the fight against cancer and that is why the use of real data, combined with machine learning techniques and help of the doctors, experts on the field, was particularly interesting.

About the intellectual property of the project, the model proposed and its implementation will be available in my GitHub (github.com/franciscocobo) account, open to anyone who would like to use it for another problem. The model is very generic and could be used in any database containing real and binary data, and having from 100 up to 1000 data samples.

10. SOCIO-ECONOMIC IMPACT

In this chapter, the impact of the project will be explained. The timetable of the work and a budget for a posterior scientific research based on the results of the thesis will also be provided.

10.1. Impact of the project

The project results will lead to a better and easier classification of patients with AML when treated in the Hospital. This will mean better, sooner and more personalized treatment for each of the patients, according to their particular characteristics. The economic impact will be a less waste of resources and time of the doctors as they will know better how to approach the treatment of the patient based on the group that belongs to. However, the biggest impact of the project is the social one, as it is a new small step in the fight against cancer and particularly a very important and still difficult to treat correctly type like AML is. The work put together by the multidisciplinary team throughout the project and the work that is still yet to do, aimed and will aim for getting interesting insights out of the data that can help in the treatment of the disease. The ethics behind the project are very important as well because of the treatment of real personal data, that was assured to be anonymized. All legal regulations previously mentioned were followed and the investigation was approved by the Ethics Committee of the Hospital.

10.2. Project planning

In the following table and Gantt Chart (using a template from [27]), the planning of the project is explained.

Task Name	Start Date	End Date	Duration (Days)	Days Complete	Days Remaining	Percent Complete
Reading of documentation	11/9/17	10/11/17	60	60,00	0,00	100%
Studying the algorithms	2/10/17	11/11/17	40	40,00	0,00	100%
Implementation of the model	11/11/17	10/1/18	60	60,00	0,00	100%
Testing the implementation	10/1/18	9/2/18	30	30,00	0,00	100%
Data curation	9/2/18	1/3/18	20	20,00	0,00	100%
Testing the model with real data	1/3/18	21/3/18	20	20,00	0,00	100%
Interesting metrics for the model	21/3/18	5/4/18	15	15,00	0,00	100%
Creation of the result document for the doctors	5/4/18	25/4/18	20	20,00	0,00	100%
Memory writting	30/4/18	9/6/18	40	40,00	0,00	100%

Table 10.1. Timetable of the project planning

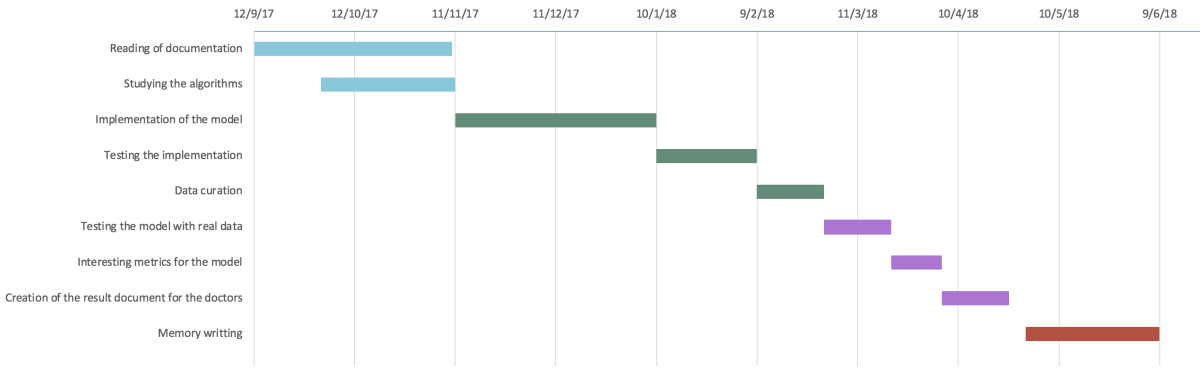


Fig. 10.1. Gantt diagram of the project

The total time taken for the development of the project was 9 months.

10.3. Project budget

As this thesis is thought as a first approach for a later wider research project, a tentative budget for a possible research project proposal that aims at continuing with this research will be provided.

The estimated time for the project to be completed will be 2 years. The research team will be formed by two people, the main researcher and the graduate engineer. The costs of the project will be the following.

The computing needs, for the machine learning nature of the project, will be provided by the research group of the university that the project will be carried out in, the Signal Processing and Learning Group (GTSA). The staff costs will be the salaries for the graduate engineer, full time, that can be extracted from the official UC3M table 10.3. The main researcher salary will be covered by the university. A personal computer will also be needed. The financial needs for trips and congresses, due to the nature of the project will also need to be covered. And lastly, it is important to include the costs for the publishing of the papers developed in the official repositories. The costs on the Hospital side are assumed to be covered separately.

These costs are summarized in the following table.

ITEM	TOTAL COST
Cost of graduate engineer	2150,34€/month x 24 months = 51.608,16 €
Personal computer	2.500 €
Trips and congresses	3.000€/year x 2 years = 6.000 €
Papers publishing	5.000 €
	65.108,16 €

Table 10.2. Budget of the research project following the thesis

Bringing the total estimated cost of the research project lasting two years to 65.108,16 euros.

COSTE MENSUAL DE CONTRATACIÓN LABORAL*, SEGÚN TABLAS PARA 2016

	Bachiller / FP2		Titulado Medio		Titulado Superior			
					MÍNIMO		MÁXIMO	
JORNADA	Salario bruto	Coste proyecto	Salario bruto	Coste proyecto	Salario bruto	Coste proyecto	Salario bruto	Coste proyecto
17,5H	910,00	1236,24	1056,35	1435,06	738,67	1003,49	1265,46	1719,13
22,5H	1170,00	1589,45	1358,17	1845,08	949,72	1290,20	1627,02	2210,32
27,5H	1430,00	1942,67	1659,98	2255,09	1160,77	1576,91	1988,58	2701,50
32,5H	1689,99	2295,87	1961,80	2665,11	1371,82	1863,63	2350,14	3192,68
Completa (37,5H)	1949,99	2649,08	2263,61	3075,13	1582,87	2150,34	2711,70	3683,86
	Coste hora=	20,18	Coste hora=	23,43	Coste hora=	16,38	Coste hora=	28,07

Table 10.3. Official UC3M personal costs table

11. CONCLUSIONS

11.1. The project

During my bachelor in Telecommunications Engineering I have learned how to extract complex information from technical books and documents. I have also become familiar with important habits like never stopping in trying to understand the concepts; or go deeper than just merely reading about something in the first source than comes up, but rather consult and compare several documents. All those skills, mixed with the phenomenal mentorship by my supervisor and my great desire in building an interesting project have led me into finishing this thesis.

The main purpose of the project was to work with an interesting medical database in order to generate new insights that could be discussed with the doctors, experts in the field, and thus create useful conclusions. The creation of this multidisciplinary team aims to have a long-term relationship that leads to several studies using data science. This bachelor thesis is a first approach, leading to some preliminary results that can be used in the future for a longer study that informs the doctor how to divide the patients in subgroups, helping in personalized medicine and treatments.

The technical part of the project was the creation of a new generative model that could describe the given database, which had both real and binary data. In order to treat that data, an Expectation Maximization algorithm for a Gaussian-Bernoulli mixture model was proposed and developed. The model and its implementation were checked with several toy examples with ground truth before applying it to the real medical database. Also, several metrics were defined in order to extract not only the division of the patients in groups but also information about how the algorithm performed the grouping or *clustering*. The dataset was analyzed from an agnostic point of view, as the team in the university did not know much about the disease studied and the variables included. The results obtained, and their analysis, were then sent to the doctors from Madrid's Hospital Gregorio Marañón to be double checked, obtaining clear and validated conclusions. As an extension, the analysis of the effect of the TPH was provided, showing the effectiveness of the treatment.

It has been discovered how important are the cytogenic risk and the type of LMA in order to determine the length of the patient survival. It is also now known that the leukocytes, the LDH, the percentage of blasts and the genetic mutation NPM1/FLT3 are very important characteristics of the patient in order to assign to which cluster must belong.

11.2. Met goals

The model proposed allows to obtain in a very clear and interesting way how the variables under study affect the different groups of patients. If they are also compared with variables external to the analysis, like the survival time of the patients (which is subsequent to the diagnosis), allows to obtain clear conclusions. Thus, the model proposed explains well the database and it is also very generic so that it can be used in different problems. Also, the effectiveness of the TPH is proved with a simple analysis carried out on the database. All these results are very useful in the fight against cancer.

11.3. Future work

As mentioned before, this thesis is thought as a first approach to a wider posterior analysis. Possible improvements for this thesis and for a similar work would be to work with a model that divided the patients in hierarchical groups that could explain better the differences among the clusters. Also, a generative model that included other distributions of data that model in a more precise way the variables studied could obtain better results.

BIBLIOGRAPHY

[1] American-Cancer-Society. (2014). What is acute myeloid leukemia?, resource accessed in may 2018, [Online]. Available: https://www.cancer.org/cancer/acute-myeloid-leukemia/about/what-is-aml.html.

[2] NHS. (2016). Acute myeloid leukaemia, resource accessed in may 2018, [Online]. Available: https://www.nhs.uk/conditions/acute-myeloid-leukaemia/.

[3] Leukaemia-Foundation. (2018). Acute myeloid leukaemia (aml), resource accessed in may 2018, [Online]. Available: https://www.leukaemia.org.au/disease-information/leukaemias/acute-myeloid-leukaemia/.

[4] MathWorks. (2018). What is deep learning?, resource accessed in may 2018, [Online]. Available: https://www.mathworks.com/discovery/deep-learning.html.

[5] C. Bishop, *Pattern Recognition and Machine Learning*. KYO, 2006.

[6] A. K. Jain, R. P. W. Duin, and J. Mao, *Statistical Pattern Recognition: A Review*. IEEE, 2000.

[7] A. Juan, J. G. Hernández, and E. Vidal, *EM initialisation for Bernoulli mixture learning*. DSIC, Universitat Politecnica de Valencia, 2004.

[8] STATA. (2015). Finite mixture models (fmms), resource accessed in may 2018, [Online]. Available: https://www.stata.com/new-in-stata/finite-mixture-models/.

[9] S. N. Srihari, *Machine Learning: Generative and Discriminative Models*. Department of Computer Science and Engineering, University at Buffalo, 2017.

[10] Wikipedia. (2018). Unsupervised learning, resource accessed in march 2018, [Online]. Available: https://en.wikipedia.org/wiki/Unsupervised_learning.

[11] D. Xiaofeng, *A joint finite mixture model for clustering genes from beta, Gaussian and Bernoulli distributed data*. University of Tampere, Finland, 2009.

[12] K. Yu, *EM Algorithm for Gaussian Mixture Model*. SJTU Speech Lab, 2013.

[13] Wikipedia. (2018). Jensen's inequality, resource accessed in march 2018, [Online]. Available: https://en.wikipedia.org/wiki/Jensen27s_inequality.

[14] NumPy-Developers. (2005). Numpy is the fundamental package for scientific computing with python, resource accessed in may 2018, [Online]. Available: http://www.numpy.org/.

[15] E. Jones, T. Oliphant, P. Peterson, *et al.* (2001). SciPy: Open source scientific tools for Python, resource accessed in may 2018, [Online]. Available: http://www.scipy.org/.

[16] Numfocus. (2010). Python data analysis library, resource accessed in may 2018, [Online]. Available: https://pandas.pydata.org/.

[17] M. Zabarauskas. (2013). Expectation-maximization algorithm for bernoulli mixture models (tutorial), resource accessed in may 2018, [Online]. Available: http://blog.manfredas.com/expectation-maximization-tutorial/.

[18] S. Srihari. (2017). Introduction to machine learning course, resource accessed in may 2018, [Online]. Available: https://cedar.buffalo.edu/~srihari/CSE574/.

[19] Y. LeCun, C. Cortes, and C. J. Burges. (2004). The nminst database of handwritten digits, resource accessed in may 2018, [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[20] J. G. Dy and C. E. Brodley, *Feature Selection for Unsupervised Learning*. Journal of Machine Learning Research, 2004.

[21] S. Hélie, *An Introduction to Model Selection: Tools and Algorithms*. Université du Québec À Montréal, 2006.

[22] J. J. Rifón, *Transplant of hemopoietic progenitors*. Servicio de Hematología y Área de Terapia Celular. Clínica Universitaria de Navarra. Pamplona., 2006.

[23] GDPR-Portal. (2018). Gdpr key changes, resource accessed in may 2018, [Online]. Available: https://www.eugdpr.org/the-regulation.html.

[24] MUNDOLODP. (2018). El rgpd no conllevará problemas a la investigación biomédica, resource accessed in may 2018, [Online]. Available: http://www.mundolopd.com/lopd/rgpd-sin-problemas-investigacion-biomedica/.

[25] Agencia-española-de-protección-de-datos. (2018). Informe: 073667/2018, resource accessed in may 2018, [Online]. Available: https://www.aepd.es/media/informes/2018-0046-investigacion-biomedica.pdf.

[26] European-Commission. (2018). 2018 reform of eu data protection rules, resource accessed in may 2018, [Online]. Available: https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en.

[27] TeamGantt. (2018). The best free gantt chart excel template, resource accessed in may 2018, [Online]. Available: https://www.teamgantt.com/free-gantt-chart-excel-template.

[28] C. Schröder and S. Rahmann, *A hybrid parameter estimation algorithm for beta mixtures and applications to methylation state classifcation*. Algorithms for Molecular Biology, 2017.

[29] C. Chan and J. McCarthy. (2017). Computational statistics in python, resource accessed in may 2018, [Online]. Available: http://people.duke.edu/~ccc14/sta-663-2017/.

ANNEX A. NOMENCLATURE

UC3M	Universidad Carlos III de Madrid
AML	Acute Myeloid Leukemia
GDPR	General Data Protection Regulation
EM	Expectation Maximization
GMM	Gaussian Mixture Model
BMM	Bernoulli Mixture Model
GBMM	Gaussian-Bernoulli Mixture Model
ML	Maximum Likelihood
PDF	Probability Density Function
BIC	Bayesian Information Criterion
LDH	Lactate Dehydrogenase
ECOG	Eastern Cooperative Oncology Group
MO	Methyl Orange
TPH	Transplant of Hemopoietic Progenitors